Pose-invariant, model-based object recognition, using linear combination of views and Bayesian statistics.

Vasileios Zografos

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of the

University of London.

Department of Computer Science

University College London

2009

I, Vasileios Zografos, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

This thesis presents an in-depth study on the problem of object recognition, and in particular the detection of 3-D objects in 2-D intensity images which may be viewed from a variety of angles. A solution to this problem remains elusive to this day, since it involves dealing with variations in geometry, photometry and viewing angle, noise, occlusions and incomplete data. This work restricts its scope to a particular kind of extrinsic variation; variation of the image due to changes in the viewpoint from which the object is seen.

A technique is proposed and developed to address this problem, which falls into the category of view-based approaches, that is, a method in which an object is represented as a collection of a small number of 2-D views, as opposed to a generation of a full 3-D model. This technique is based on the theoretical observation that the geometry of the set of possible images of an object undergoing 3-D rigid transformations and scaling may, under most imaging conditions, be represented by a linear combination of a small number of 2-D views of that object. It is therefore possible to synthesise a novel image of an object given at least two existing and dissimilar views of the object, and a set of linear coefficients that determine how these views are to be combined in order to synthesise the new image.

The method works in conjunction with a powerful optimization algorithm, to search and recover the optimal linear combination coefficients that will synthesize a novel image, which is as similar as possible to the target, scene view. If the similarity between the synthesized and the target images is above some threshold, then an object is determined to be present in the scene and its location and pose are defined, in part, by the coefficients. The key benefits of using this technique is that because it works directly with pixel values, it avoids the need for problematic, low-level feature extraction and solution of the correspondence problem. As a result, a linear combination of views (LCV) model is easy to construct and use, since it only requires a small number of stored, 2-D views of the object in question, and the selection of a few landmark points on the object, the process which is easily carried out during the off-line, model building stage. In addition, this method is general enough to be applied across a variety of recognition problems and different types of objects.

The development and application of this method is initially explored looking at two-dimensional problems, and then extending the same principles to 3-D. Additionally, the method is evaluated across synthetic and real-image datasets, containing variations in the objects' identity and pose. Future work on possible extensions to incorporate a foreground/background model and lighting variations of the pixels are examined.

Acknowledgements

To my family for their constant support, sacrifice and belief and to Prof. Bernard Buxton for his help, encouragement and valuable guidance during all these years.

Vasileios Zografos,

London, September, 2009

Contents

1 Introduction **14**

1.1 Problem statement 14

1.2 Aim 16

1.3 Objectives 16

1.4 Main hypothesis statement 17

1.4.1 Hypothesis 1 18

1.4.2 Hypothesis 2 19

1.4.3 Hypothesis 3 20

1.5 The approach 20

1.5.1 Why a Bayesian approach? 22

1.6 The contributions made in this thesis 23

1.7 The significance of this work 25

1.8 Papers resulting from this thesis 26

1.9 Definitions 26

1.10 Abbreviations 27

1.11 Structure of this dissertation 27

2 Related work **29**

2.1 Choice of coordinate system 29

2.1.1 Viewer-centred approach 29

2.1.2 Object-centred approach 30

2.2 Choice of strategy: features vs templates 31

2.2.1 Feature-based approach 31

2.2.2 Image-based approach 33

2.3 Choice of model representation 35

2.3.1 Feature points 35

2.3.2 Curves 35

2.3.3 Orthogonal basis 36

2.3.4 Image templates 36

2.4 Deformable template models 37

2.4.1 Free-form models . . . 37
2.4.2 Parametric deformable models . . . 38
2.5 Support vector machines . . . 40
2.6 Optimisation . . . 41
2.6.1 Local methods . . . 41
2.6.2 Global methods . . . 45
2.7 Active Appearance Models . . . 50

3 Background theory **53**
3.1 Single view geometry . . . 53
3.2 Multi-view geometry . . . 56
3.3 Linear combination of views . . . 57

4 2-D object recognition **59**
4.1 Model representation . . . 59
4.2 Parametric transformations . . . 61
4.3 Probabilistic constraints . . . 65
4.4 Objective function . . . 69
4.4.1 The scale transformation . . . 71
4.5 Experimental results . . . 77
4.6 Basic foreground/background modelling . . . 82
4.7 Summary . . . 86

5 3-D object recognition **88**
5.1 The recognition system: Rigid objects . . . 88
5.1.1 Modelling . . . 90
5.1.2 Image synthesis . . . 92
5.1.3 Matching . . . 94
5.1.4 Coefficient variation . . . 95
5.2 Bayesian model . . . 98
5.2.1 Likelihood . . . 99
5.2.2 Prior . . . 99
5.2.3 Posterior . . . 100
5.3 Experimental results . . . 102
5.3.1 Markov-Chain Monte-Carlo . . . 106
5.4 Summary . . . 110

6 Optimisation strategy **112**
6.1 2-D test functions . . . 112
6.2 Real-image template matching . . . 113

6.3 Experiments: methods and results . . . 116
6.3.1 Set 1: 2-D test functions . . . 116
6.3.2 Set 2: Real-image template matching . . . 120
6.3.3 Hybrid approach . . . 125
6.4 Summary . . . 127

7 Experiments and evaluation 129
7.1 Image datasets . . . 129
7.1.1 Database 1: Synthetic dataset . . . 130
7.1.2 Database 2: COIL-20 . . . 130
7.1.3 Database 3: Yale Face Database B . . . 132
7.2 Training . . . 134
7.3 Proposed experiments . . . 135
7.3.1 K-fold cross validation . . . 136
7.3.2 Experiments on database 1 (Synthetic database) . . . 136
7.3.3 Experiments on database 2 (COIL-20 database) . . . 137
7.3.4 Experiments on database 3 (Yale face database B) . . . 137
7.3.5 Comparison with AAMs . . . 138
7.4 Results . . . 139
7.4.1 Database 1 . . . 139
7.4.2 Database 2 . . . 145
7.4.3 Database 3 . . . 159
7.4.4 Noise . . . 171
7.4.5 Occlusion . . . 178
7.4.6 Expression . . . 187
7.4.7 Rotation about a horizontal axis . . . 188
7.4.8 Illumination variation . . . 194
7.5 Summary . . . 197

8 Conclusions 199
8.1 Research summary . . . 199
8.2 Critical evaluation - Remarks . . . 202
8.2.1 Hypothesis 1 . . . 203
8.2.2 Hypothesis 2 . . . 204
8.2.3 Hypothesis 3 . . . 205
8.2.4 Main hypothesis statement . . . 206
8.3 Limitations and future work . . . 207

Appendices 212

A Algorithms **212**

B Exploratory data analysis techniques **214**

Bibliography **216**

List of Figures

2.1 The classification of deformable template models in recent research.) 38
2.2 Common optimisation methods traditionally used in computer vision. 42
2.3 Possible simplex moves after the worst point (W) is identified and rejected. 43
2.4 Comparison between a fixed (a) and a reducing-step (b) restarting simplex. 44
2.5 The two examined step reduction schedules. 44
2.6 Reproduction children in a typical genetic algorithm. 47
2.7 2-D examples of the SOMA algorithm. 50

3.1 Pinhole camera geometry showing the projection of a point P to the image plane Π. . . . 53
3.2 Common approximations to the perspective camera. 55

4.1 surfaces and curves composing the affine transformation. 61
4.2 The sinusoidal wave (a) and its deformation effects (c) on a 2-D shape (b). 64
4.3 Error function appearance for a match located at successively higher scale values. 67
4.4 The mixture model (bold line) for the distribution of the shear parameter ϕ. 69
4.5 Comparison of the Huber and smooth Huber norms. 71
4.6 A model of the spherical imaged object under a perspective camera model. 72
4.7 Distance vs Scale theoretical process model. 73
4.8 Typical captured image sample (a) and the prototype template (b). 74
4.9 Sampled histograms of the scale from the first (a) and second (b) experiments. 74
4.10 Lognormal probability plot (a), estimated pdf (b) and cdf (c) plots from sample data. . . 77
4.11 Video sequence results. (a) prob. plot, (b) pdf plot, (c) cdf plot and (d) lag plot. 78
4.12 Experiments on real images with randomly transformed templates. 80
4.13 (a) Manually adjusted scale space and (b) comparison between Euclidean distances. . . 81
4.14 The effects of the lognormal prior on the scale parameter surface. 81
4.15 Simple matching examples and error surfaces. 86

5.1 An outline of the proposed recognition system. 89
5.2 Modelling steps: (a) basis view (b) landmark points and (c) triangulation 92
5.3 The variation of the 10 coefficients for horizontal rotation. 96
5.4 Negative log-posterior plots for 3 of the coefficients. 101
5.5 Synthetic data used for the testing of the LCV object recognition approach 102

5.6 Two synthesised examples at the chosen thresholds. (a) c.c=0.966 and (b) E_B=108 . . . 103
5.7 The diversity of the coefficients from the 100 tests with good-initialisation. 104
5.8 The average optimisation behaviour of the 3 examples. 104
5.9 Comparison between the two measures for the good-initialisation and Bayesian tests. . . 106
5.10 The identified clusters before (b) and after (a) thinning-out the sample. 107
5.11 The diversity of the 10 coefficients before (a) and after (b) the thinning-out. 109

6.1 The five 2-dimensional test functions. 114
6.2 Commonly encountered datasets and their corresponding translation error spaces. 116
6.3 Comparison between parameter displacement and error response. 121
6.4 The diversity of the rotation angle in the first dataset using GA (a), DE (b) and SOMA (c). 123
6.5 The average converged test runs for all the 3 datasets. 124
6.6 Plots comparing the hybrid approach and the SOMA method for the 3 datasets. 127

7.1 Typical samples from the synthetic database at various rotation angles (hor.,vert.) 129
7.2 Synthetic database sample, showing the landmark points and Delaunay triangulation. . . 130
7.3 Synthetic samples with different expression, noise and occlusion levels. 131
7.4 Image samples from the COIL-20 database. 131
7.5 A typical sample from the COIL-20 database with chosen landmarks visible. 131
7.6 All the 10 individuals in the Yale face database B. 132
7.7 (a) sample background in the Yale database and (b) sample landmark points. 133
7.8 All the different pose angles in the Yale face database B. 133
7.9 Example of basis views training errors for the synthetic dataset. 135
7.10 COIL-20 sample with superimposed AAM in typical starting position. 138
7.11 RMSE and MAE plots using cross-correlation. 142
7.12 Average cross-correlation plot (mode of sample). 142
7.13 Full cross-correlation data histogram for the pose variation. 143
7.14 Average back projection error (mode of sample). 143
7.15 Full back projection data histogram for the pose variation. 143
7.16 Diversity of mean coefficients for pose variation. 144
7.17 Acceptance $\%$ of test results for different thresholds. 144
7.18 Average cross-correlation plot (mode of sample) using AAMs. 145
7.19 Average back projection error (mode of sample) using AAMs. 146
7.20 RMSE and MAE plots for cross-correlation, using AAMs. 146
7.21 RMSE model$\times$object array for the frontal pose using LCV. 148
7.22 CC model$\times$object array for the frontal pose using LCV. 148
7.23 Acceptance ratio for Model$\times$object at the frontal pose using cc. 149
7.24 Acceptance ratio for Model$\times$object at the frontal pose using BP. 150
7.25 BP model$\times$object array for the frontal pose using LCV. 151

7.26 RMSE model×object array for the frontal pose using AAMs. 152
7.27 CC model×object array for the frontal pose using AAMs. 153
7.28 BP model×object array for the frontal pose using AAMs. 153
7.29 Acceptance ratio for Model×object at the frontal pose using AAM. 154
7.30 RMSE pose variation plot for the COIL-20 database, using LCV. 155
7.31 CC pose variation plot for the COIL-20 database, using LCV. 156
7.32 BP pose variation plot for the COIL-20 database, using LCV. 156
7.33 Acceptance performance surface plot for COIL-20 database, using LCV. 157
7.34 RMSE pose variation plot for the COIL-20 database, using AAMs. 158
7.35 CC pose variation plot for the COIL-20 database, using AAMs. 158
7.36 BP pose variation plot for the COIL-20 database, using AAMs. 158
7.37 Acceptance performance surface plot for COIL-20 database, using AAMs. 159
7.38 RMSE model×object array for the frontal pose, using LCV. 160
7.39 CC model×object array for the frontal pose, using LCV. 161
7.40 BP model×object array for the frontal pose, using LCV. 162
7.41 Acceptance performance surface plot for Yale B database, using LCV. 162
7.42 RMSE model×object array for the frontal pose, using AAMs. 163
7.43 CC model×object array for the frontal pose, using AAMs. 164
7.44 BP model×object array for the frontal pose, using AAMs. 164
7.45 Acceptance performance surface plot for Yale B database, using AAMs. 165
7.46 RMSE object=model×pose array, using LCV. 166
7.47 CC object=model×pose array, using LCV. 167
7.48 BP object=model×pose array, using LCV. 168
7.49 Acceptance performance surface plot for Yale B database, using LCV. 168
7.50 RMSE object=model×pose array, using AAMs. 168
7.51 CC object=model×pose array, using AAMs. 169
7.52 BP object=model×pose array, using AAMs. 170
7.53 Acceptance performance surface plot for Yale B database, using AAMs. 170
7.54 Synthetic database samples with different amount of random noise. 172
7.55 RMSE and MAE plots for moderate noise case. 172
7.56 Average cross-correlation plot (mode of sample) for moderate noisy case. 173
7.57 Average BP plot (mode of sample) for moderate noisy case. 173
7.58 Recognition rates comparison using CC and BP score thresholds. 173
7.59 RMSE and MAE plots for moderately noisy case using AAMs. 174
7.60 Average cross-correlation plot (mode of sample) for moderately noisy case. 174
7.61 Average BP plot (mode of sample) for moderately noisy case. 175
7.62 Recognition rates comparison between LCV and AAM methods. 175
7.63 RMSE and MAE plots for extensively noisy case. 176

7.64 Average cross-correlation plot (mode of sample) for extensively noisy case. 177
7.65 Average BP plot (mode of sample) for extensively noisy case. 177
7.66 Recognition rate comparison using CC and BP score thresholds. 177
7.67 RMSE and MAE plots for extensively noisy case, using AAMs. 179
7.68 Average cross-correlation plot (mode of sample) for extensively noisy case. 179
7.69 Average BP plot (mode of sample) for extensively noisy case. 179
7.70 Recognition rates comparison using CC and BP error thresholds. 180
7.71 Synthetic database samples with different amount of random occlusion. 180
7.72 RMSE and MAE plots with 20% occlusion. 181
7.73 Average cross-correlation plot (mode of sample) with 20% occlusion. 182
7.74 Average BP plot (mode of sample) with 20% occlusion. 182
7.75 Recognition rates comparison using CC and BP score thresholds. 182
7.76 RMSE and MAE plots with 20% occlusion using AAMs. 183
7.77 Average cross-correlation plot (mode of sample) using AAMs. 184
7.78 Average BP plot (mode of sample) using AAMs. 184
7.79 Total data histogram for 20% occlusion, using AAMs. 184
7.80 Recognition rates comparison using CC and BP score thresholds. 185
7.81 RMSE and MAE plots with 40% occlusion. 186
7.82 Average CC and BP plots (mode of sample). 186
7.83 Average CC and BP plots (mode of sample). 186
7.84 Recognition rates comparison using CC and BP score thresholds. 187
7.85 Average CC and BP plots (mode of sample). 187
7.86 Average CC comparison for unmodelled expressions. 189
7.87 Average BP comparison for unmodelled expressions. 189
7.88 Acceptance comparison for unmodelled expressions. 189
7.89 RMSE and MAE plots for horizontal rotation. 190
7.90 Average CC and BP responses for horizontal rotation. 190
7.91 Average acceptance comparison for CC and BP score thresholds. 191
7.92 Diversity of mean coefficients. 191
7.93 RMSE vs MAE plot for horizontal rotation using AAMs. 193
7.94 Average CC and BP responses for rotation about a vertical axis using AAMs. 193
7.95 Recognition comparison between LCV and AAMs. 193
7.96 Position of illumination sources relative to the camera. 195
7.97 CC response under non-linear illumination variation. 195
7.98 BP error response under non-linear illumination variation. 195

List of Tables

4.1 Quantitative results for the lognormal distribution. 78
4.2 Comparison between actual and estimated transformation values from Fig. 4.12(d),(e). . 81

5.1 Object recognition results for the 3 different cases. 104
5.2 The centres of the five identified clusters with their associated c.corr. values. 108
5.3 The results from the numerical tests on the drawn sample. 109

6.1 The test results for the 5 functions using a reducing-step restarting simplex. 117
6.2 The test results for the 5 functions using a pattern search algorithm. 118
6.3 The test results for the 5 functions using a genetic algorithm. 119
6.4 The test results for the 5 functions using DE. 119
6.5 The test results for the 5 functions using SOMA. 120
6.6 Comparative results from the 3 datasets using all the algorithms. 123
6.7 The results of the hybrid and SOMA tests at 6000 and 20000 FEs. 127

7.1 Acceptance results for pose variation at different thresholds. 144
7.2 Acceptance results for pose variation at different thresholds, using AAMs. 146

Chapter 1

Introduction

Object recognition is one of the most important and basic problems in computer vision. It may broadly be defined as the task of recognizing and locating objects from the real world in a representation (image) of the world, using object models that are known a priori. In this scenario, the system is given image data that contain foreground (areas of interest) and background objects, and a set of labels that correspond to a set of models known to the system. The object recognition system must then assign the correct labels to the appropriate regions in the image. Object recognition has been studied extensively in the past, resulting in a number of publications and a variety of different approaches [Jain et al. (1998); Pope (1994); Yang et al. (2002); Besl and Jain (1985)] aiming to solve different aspects of the problem.

Nevertheless, accurate, robust and efficient solutions remain elusive to this day because of the inherent difficulties when dealing in particular with 3-D objects that may be seen from a variety of viewpoints. Variations in geometry, photometry and viewing angle, noise, occlusions and incomplete data are some of the problems with which object recognition systems are faced. In all cases, prior information about the object is available in the form of a model which is matched to the object(s) in the input image, in some kind of optimisation scheme often expressed as an "energy" minimisation.

This work examines a view based approach in which 2-dimensional view-centred representations of 3-dimensional objects, called aspects, characteristic views [Koenderink and van Doorn (1979)] or basis views [Ullman and Basri (1991)] are used. Such methods have recently become quite popular because, in principle, they are applicable in many areas and easy to implement, since they avoid generating and storing a full 3-D model. In addition, there is evidence to suggest that view-based representations may be used by the human visual system for object recognition [Bülthoff and Edelman (1992); Tarr and Büllthoff (1998); Tarr et al. (1998)].

1.1 Problem statement

Any 3-D object may be represented as one or more images taken from different viewpoints. In most object recognition scenarios the object of interest is at a viewing distance that gives a clear view of the object as a whole with sufficient detail visible to render it distinctive. In such a scenario, the depth variation across the object of interest is usually sufficiently small in comparison to its distance from the camera that the perspective projection may be well-approximated by an affine projection. In a view-

based object recognition approach, or in other words, the problem of recognising a flat object from a single 2-D image may then be formulated as follows:

> *Suppose we are given a prototype template function F_0, a "target" scene image function I and a transformation T that transforms the template as: $F = TF_0$. F, F_0, I are all discrete functions that may represent feature vectors in a **feature-based** approach or pixel intensities or colour attributes in an **image-based** approach. The goal of object recognition is to minimise the expression:*
>
> $$\hat{p} = \underset{T}{\operatorname{argmin}}\, g(I(x), F(x)), \tag{1.1}$$
>
> *with respect to the transformation T, defined by a set of parameters ξ. $g(...)$ is a matching metric giving rise either to a dissimilarity or similarity score (e.g. Euclidean distance or cross-correlation coefficient), both of which may be cast as criteria to be minimised. If the minimum at $\hat{p}$ is less than or equal to some threshold τ, then we say we have a match, attach the appropriate labels to the region of the image function I corresponding to the model defining the object of interest in the template F_0, and say that the object in the image has been recognised.*

The main difficulty that arises in the above formulation is the determination of the transformation parameters ξ that minimise (1.1) since solving for ξ depends on the type of transformation T. There is a closed form solution of (1.1) when T is an affine transformation acting on point features and a sum of squared error metric is used, but this requires solution to both the feature extraction and correspondence problems, both of which are not usually straightforward as we shall see later. If on the other hand we use pixel values, then there is no closed-form solution and the problem becomes one typical of template matching. In this case, and for complicated transformations T, minimisation of of (1.1) is a non-linear, non-invertible process that requires a different approach to its solution. Determination of the optimal coefficients ξ of the transformation T for the image-based case when pixel values are used, is one of the main focus areas of this research.

Once this problem has been resolved for a single 2-D view, the next step is to make use of the view-based approach. This involves using more than one representative view of the object at the same time. In this approach, 3-dimensional objects are represented by methods based on a combination of 2-D images or line drawings. [Ullman and Basri (1991)] developed this approach for representing primarily rigid objects by using a linear combination of line drawings or edge maps, often known as a *linear combination of views* or LCV for short. Following the initial work of [Ullman and Basri (1991)], others have taken this concept further to the combination of images themselves [Koufakis and Buxton (1998b); Hansard and Buxton (2000b); Peters and von der Malsburg (2001)]. These techniques produce very good, realistic looking representations of an image, but are limited to rigid objects and break down when used for models that can undergo non-rigid deformations. Recently, Dias [Dias (2004)] has addressed this problem and extended the LCV technique to work for objects that can change shape. His

method however, is a feature-based approach that does not take into consideration pixel intensity or colour information, but instead relies on the existence of known landmark points around prominent features both in the model and in the target image.

In summary, determination of the optimal transformation parameters ξ and extension of (1.1) to utilise LCV representations, in order to build a system able to recognise a rigid 3-D object from its 2-D views, using pixel intensity information alone, are the primary areas of research addressed in this work.

1.2 Aim

The main aim of this research has been to carry out a new study on the area of object recognition via model-based, multi-view template matching and its associated problems and deficiencies. More specifically, we focused on examination of the linear combination of views theory and its extension to more complicated objects and, in particular, using image pixel values rather than simplistic line drawings or point features.

This is in fact the principal hypothesis on which this thesis is based, namely that such an extension is possible and can lead to a successful object recognition and localisation scheme. The intention is therefore to propose a new strategy for solving a number of problems associated with this pixel-based, LCV approach to object recognition and extraction, such as the problem of localisation and matching, template search and optimisation in a high dimensional space, and image variation due to changes in the viewpoint from which the object is seen. Each of these problems is addressed in more detail in later sections.

1.3 Objectives

In order to meet the main aim of this research of demonstrating that a successful pixel-based, LCV object recognition scheme can be developed, a system is implemented that will be characterised by the extent to which it fulfils the following objectives:

- Automatic detection and classification of the modelled object(s) in image data from viewing directions within or close to the set of basis views.
- Characterisation of an object via a small number of basis views.
- Ability to handle *sufficiently complicated* real-world objects without giving preference to a specific class of shapes (e.g. curved or planar surfaces).
- Ability to function with a certain amount of *noise* in the data, without an un-due, disproportionate degradation in performance.
- Ability to handle arbitrary combinations of a relatively *large number* of objects in a variety of orientations and locations without being overly sensitive to small amounts of occlusion.

In addition to the above the system should be able to perform within some error limits. More specifically, it should have a low tolerance for *miss errors* (when an object's presence is not detected),

false alarm errors (when the presence of an object is indicated even though it is not present in the input target image) and *localisation errors* (when an object's presence in the target image is correctly determined but its identified location is incorrect).

It is also to be noted that incorporating the effect of occlusions is almost but not entirely straightforward because of the need, in principle, for a correct statistical approach to estimate the likelihood of a particular object's presence by using data from over the whole of the target image. Occluding objects thus naturally become part of the recognition scheme along with the image background and they must be known a-priori or modelled in some manner. For the most part, we will usually assume that the background is known a-priori though we note the possibility of modelling it statistically as characteristic of say, natural or man-made scenes [Huang and Mumford (1999); Grenander and Srivastava (2001); Sullivan et al. (1999)]. In principle, of course, the whole image both foreground and background could and should be modelled by the same LCV methods. This would take us beyond the scope of the present work, but given that an occluding object is necessarily in front of the foreground object of interest, such an approach would be most appropriate. Other ways of modelling of occluding objects can be problematic. This thesis therefore includes only a small number of experiments on synthetic data that although they may not be rigorously valid, help to demonstrate the performance of the method in the presence of a limited amount of occlusion. There is also the case of self-occlusions when the modelled object is non-convex, which although are not specifically tested in this thesis, could also be taken into account in the LCV approach by utilising the affine depth as in [Hansard and Buxton (2000b)]. Since [Hansard and Buxton (2000b)] shows that the appearance of a self-occluding object can be modelled well in the LCV approach, there is little reason to suppose that an extension of our object recognition scheme to cover such cases would not work.

1.4 Main hypothesis statement

The main hypothesis underlying this research may be given as follows:

> *A successful pixel-based scheme can be developed and implemented as a solution to the object recognition problem by integration of the linear combination of views technique (LCV) with a view-based object recognition methodology and used to build a framework for the recognition of three-dimensional, rigid objects under a variety of configurations, using a small number of images taken from different viewpoints.*

There are a number of words and phrases in the above that require further clarification. These are listed below:

- **successful**: The method or 'scheme' must be shown to work over a set of test data to a useful level of performance in particular for the recognition error rates and location accuracy as indicated in section 1.1. Synthetic data will be used for 'closed-loop' controlled experiments and widely available image databases used for more realistic tests.

- **pixel-based**: The input data pertaining to the target image (or images) in which the presence or

absence of the object (or objects) of interest is to be determined consist solely of the image pixel-values or attributes. No online pre-processing of the target image data, in particular for feature extraction, is assumed and evaluation of recognition hypotheses is carried out by reference to the target image pixel-values.

- **view-based**: Objects are to be represented by a finite (usually a small) number of images or "views" of themselves. These views or *basis images*, are to be taken under good conditions, i.e. at an appropriate resolution from a distance that allows reasonable detail on the object to be visible under affine imaging conditions, with the whole object in view, under typical illumination that does not create artefacts and is bright enough to enable appropriate surface texture and colour to be apparent.

- **object recognition problem**: The object recognition problem as defined in section 1.1.

- **framework**: An approach to object recognition based on theory and implemented in a systematic manner so that it can be followed and utilised in subsequent work by others.

- **three-dimensional rigid objects**: 3-D objects (i.e. ones that are not flat) that do not change their form in 3-D, but whose apparent shape in an image may change owing to a change of viewpoint.

- **variety of configurations**: Images taken while the camera or object is rotated about an arbitrary axis in space. Rotation about axes perpendicular to the line of sight are of most interest as they reveal the 3-dimensional nature of an object. However, this does not exclude rotations about the line of sight, also known as image-plane rotations. Such image-plane rotations may be modelled by an equation such as (4.9) as we shall see later on, which is equivalent to the LCV method using a single basis view.

- **integration**: Combination of the view-based object recognition solution with the LCV method in order to build a single unified framework.

1.4.1 Hypothesis 1

It is possible to synthesise a novel view of an object and match it to a target image of that object. A good matching score will indicate that the object is present in the scene and, barring the unlikely or deliberate presence of fakes, that it has been located accurately. The object's pose is represented by the LCV coefficients or parameters that give the best match.

This sub-hypothesis asserts that, as is known from previous work, realistic-looking images of novel views of an object can be created from a combination of a small number of basis views. Below we list words or phrases in the above, first sub-hypothesis that require clarification:

- **synthesise**: Creation of a new image of an object by linearly combining other images (usually two) of that object taken from nearby, but otherwise arbitrary viewpoints. First the geometry of the new image of the object is determined from a number of landmark points and by solving the

LCV equations, and then its appearance (colour, texture and so on) is synthesised using a series of piecewise affine warps.

- **novel view**: A view that is not in the modelling or training data set.
- **match**: A comparison between a scene and a model image that results in a good matching score, using either a similarity or dissimilarity measure. As a result the parameters of the target image object can be determined from the matched model.
- **target image**: An input image to our system in which a specific object that needs to be detected and located may exist in an arbitrary configuration. Usually, and for the purpose of this thesis, such configurations are typically the set of 3-D rigid deformations.
- **good matching score**: A matching score obtained from a predefined matching function between a model and a target image. The score is usually compared to a predetermined threshold. A value sufficiently higher or lower that the threshold (depending on whether we are using a similarity or dissimilarity matching function respectively) will indicate a high probability of a good match of the correct model to the object.
- **pose**: Model parameters associated with the extrinsic degrees of freedom of the object representing as far as possible from the available image information its position and orientation in space relative to the camera (or other frame of reference) respectively.
- **LCV coefficients**: The coefficients of the linear combination of views equations that determine (to the extent possible under affine imaging) the pose of the object in question.

1.4.2 Hypothesis 2

The introduction of prior probability distributions in the template deformation process, based on previous knowledge of the underlying image generation process and imaging conditions, can improve the accuracy and speed of the recovery of the model parameters from an image of a rigid, 3-D object.

This sub-hypothesis asserts that the imaging process and conditions can be used to predict the parameters determining the form of the model template to be matched to the foreground of the target image. Again, there are a number of words and phrases that require further explanation. These are:

- **prior distribution**: A parametric probability density function that represents our existing knowledge about the data (i.e. the process that generated the data), which is typically used in a Bayesian framework to bias the possible values of the parameters in order to avoid invalid solutions and/or guide a solution toward a specific range of values.
- **template deformation process**: Since an object may be viewed from a range of orientations, its shape in the target image will vary. The shape of the model template that is to be matched must also correspondingly vary. This is referred to as 'the template deformation process'.

- **previous knowledge**: This means that we have some scientific knowledge about the processes that generated the data. Such knowledge can be implied from the fact that object recognition is being attempted and that the object of interest must therefore appear in the target image at sufficient size and with sufficient detail visible. Ultimately such information constraining the range of possible parameter values can be expressed via a probabilistic model defined for example by a typical value or mean and the standard deviation. In practice, univariate, Gaussian distributions will be used - i.e. it will be assumed the parameters are normally distributed and correlations between them will be ignored.

- **imaging conditions**: The various properties of a scene, such as camera parameters, lighting configuration, noise and so on.

1.4.3 Hypothesis 3

> *Recovery of the optimal LCV coefficients usually requires in principle exhaustive search of the large solution space. By using an appropriate optimisation algorithm we can efficiently recover the optimal set of coefficients and thus recognise the object in the scene.*

This hypothesis reflects the fact that, as noted in section 1.1, the optimisation problem defined by equation (1.1) is, in general, complicated and non-linear and may be expected, unless the scene is very simple, to have local optima in addition to the desired global optimum of the correct, best match. Words or phrases that require further clarification are listed below:

- **in principle exhaustive**: In this case we are referring to a systematic search of the parameter space that is able to guarantee that a globally optimum solution (if one exists) is found. We cannot rule out the possibility that for simple scenes (and therefore models) the optimisation problem may be convex and therefore sometimes soluble without an exhaustive search, but in general this will not be the case in typical object recognition scenarios. We say 'in principle' because such a procedure in general is infeasible.

- **large**: The parameter space can span up to 10 dimensions depending on the use of multi-view constraints. Obviously searching such a large space exhaustively is not practical.

- **efficiently**: The desirable property of the algorithm used to solve the optimisation so that recovery of a near-optimal solution within feasible time and computation (determined as the number of function evaluations) budgets is possible.

- **optimal**: Optimal in terms of a predetermined threshold which allows us to be confident that the solution found within a given time and computation budget is close enough to a possible global optimum.

1.5 The approach

The approach presented in this thesis for solving the object recognition problem as defined in section 1.1 falls within the framework of deformable template matching algorithms where we are looking for the

transformation that maps a model to an image. In this setting, a function often from physical analogies referred to as an energy function associates a cost with each potential transformation of the model. It is desirable to find the transformation with the lowest cost below a suitable threshold.

Typically, this energy or cost function has a twofold purpose. First it attracts the deforming template toward salient image regions. Second it biases against large or otherwise undesirable deformations of the template. Since the number of possible transformations may be very large (recall the remarks above about a large, possibly 10-dimensional parameter space), it is essential to be able to search the space efficiently and guide the process toward promising regions where good solutions may lie. This is best achieved by exploiting all available prior information about the object, the scene and the imaging process. The use of a Bayesian framework combined with a powerful optimisation algorithm can achieve this purpose.

We based our approach for solving the aforementioned problem, first for a single view and later for multiple images, on the work by [Jain et al. (1996) and Bebis et al. (2002)]. These works combine a simple model of an object, a set of parametric transformations that act upon the model, each of which has an associated penalising probability distribution, and an optimisation algorithm that will recover the appropriate transformation parameters that will most closely enable the model to match with the object in the scene.

In our work, the first component, the object model, is a rectangular bitmap image (or images in the multi-view case) that contains grey-scale (or colour) pixel information of the object's contour and intensity without any additional background data. In the single view scenario (2-D objects) this bitmap may be the result of training on a number of images of the object so that it represents the most likely image appearance. For the multiple view case (3-D objects) the images are chosen so that they represent the object from different viewpoints, each containing as much information about the object as possible, since this will aid in the synthesis of the novel view and minimise any regions of missing or incomplete data on the object. Care must also be taken not to choose a very wide angle between the views, so that they do not belong to different aspects of the object, as this can lead to self-occlusions and missing data during synthesis.

The next component is the set of probabilistic transformations. These are typically learnt from appropriate training examples or empirically chosen. They combine a set of parametric transformations that deform the model with probability distributions defined on those transformations that restrict the choices of possible deformed models. The transformations we are currently considering include the 3-D rigid transformations in the multi-view case as defined by the LCV equations (3.14) and a 2-D subset in the single view case which are equivalent to a global 2-D affine transform on all the pixels in the image. Furthermore, and only for the single view case, we experimented with the addition of a local quadratic deformation designed to deal with any small non-linear effects generated during the image formation process.

The probability distributions associated with the transformations serve as a means of restricting these transformations. This can help to avoid large deformations that produce similarly substantial devi-

ations from the initial template since it is logical to assume that the model exemplifies a likely, *generic* view of the object. Furthermore, they help to avoid trivial solutions for the transformation parameters - parameter values that may minimise the energy function but produce an uninteresting result (e.g. collapse the model into a single point or line). Finally, we may also use the distributions deliberately to steer the solution away from what is previously known and guide the solution to regions of the energy surface to which it may be difficult otherwise to converge, or even just in order to investigate a wider range of possible solutions. These distributions are usually encoded as the prior distributions in a Bayesian formulation.

Our method differs from that of [Jain et al. (1996) and Bebis et al. (2002)] first as we are using pixel intensity information without the need to extract features from the target image or solve the correspondence problem. Also, we use different distributions both in the single view and multiple view cases and do not assume that all transformations are equally likely. Additionally, the likelihood function we used that expresses the probability of observing the input image given a deformed model with specific transformation parameters is based on different error metrics with which we have extensively experimented. Finally, for the recovery of the optimal transformation parameters we are using a hybrid optimisation approach that combines a recent evolutionary algorithm with a local deterministic method. This algorithm is able to produce very good results within a pre-allocated optimisation budget and without the need for strict initialisation close to the location of the desired global minimum.

A Bayesian formulation which combines this prior knowledge together with information from the input image expressed as the likelihood is therefore used in order to find a match between the image and the model. This combination of the prior and likelihood is realised in the posterior probability, a maximum of which (or equivalently a minimum of its negative logarithm) may indicate a possible match.

1.5.1 Why a Bayesian approach?

We have decided to use a Bayesian approach because tasks such as object localisation and recognition offer themselves as ideal situations for statistical inference. Such tasks are often faced with situations where only very limited and noisy data is available and, in addition, we may not be able to define an exact model to apply to this data, especially in the presence of complicated information in the background. If the data alone is unable to provide a unique solution to the problem it follows that reliable declarations about the parameters of the model (i.e. pose, location, scale and so on) cannot be made and that, in a purely data-drivel approach, the image may be well explained by a set of parameters that are, in practice, completely unrealistic.

Instead, by utilising Bayes priors we can ensure we get close enough to the correct solution with a reasonable set of model parameter values by making assumptions about these parameters based on logical reasoning from our expectation (prior knowledge) combined with observation evidence (likelihood) from the data. In our object recognition framework, Bayes' rule may be written as as:

$$P(\xi|I) = \frac{P(I|\xi)P(\xi)}{P(I)} \tag{1.2}$$

General information about the model parameters ξ is encoded in the prior probability distributions $P(\xi)$ of the transformation parameters ξ. These distributions represent our certainty about a situation *before* the data is observed. The likelihood of observing the image I given a set of parameters ξ is encoded in $P(I|\xi)$. This usually reflects noise processes that would cause the target image to deviate in detail from the model, but in our approach we must also allow for the possibility of gross errors when the model is incorrectly located or the wrong model has been selected. From the product of the likelihood and the prior probabilities we can calculate the posterior probability $P(\xi|I)$ which represents our certainty that we have explained the observed, target image I. We usually require a single model configuration to be presented as the most probable explanation. A typical choice is that for which the posterior probability is maximal (known as the maximum a-posteriori or MAP solution).

This is the main reason why probability theory and in particular Bayes' rule are appropriate tools for these kind of tasks. There are of course alternative theories that can provide similar probabilistic inference mechanisms such as the maximum likelihood (ML) solution (see [Sebe and Lew (2001, 2002); Olson (2002)]). ML tries to find a match using only the likelihood information of an event. According to [Jaynes (2003)], a model defined solely on the likelihood is incomplete, but defines only a parametric space, the maximum of which indicates a good match between model and data. By introducing the prior probability, we can incorporate information about the likely values of the model parameters that can help guide the result toward a preferred solution. Since the MAP solution differs from the ML solution only in the existence and use of the prior, it means that choosing an appropriate prior is one of the most critical aspects for the effectiveness of the MAP approach.

It is useful to note here that there are two interpretations for the prior in Bayesian theory. In the first, the "objective view", the prior represents knowledge acquired in a previous experiment. In other words, it might be (and usually is) the posterior probability of the previous experiment. In such cases, we start our inference by using an uninformative prior (such as the uniform distribution) and we iteratively update our knowledge (i.e. $P_m(\xi) = P_{m-1}(\xi|I_{m-1})$ where m is the iteration number and I_{m-1} the information available after $m-1$ iterations) as the new data is made available. In the second, the "subjective view", there is no data from previous experiments, but instead the data is made available simultaneously and not sequentially as in the previous case. If we have some general information about the parameters ξ we can chose an appropriate prior distribution $P(\xi)$ that reflects this knowledge in order to restrict ξ so that the posterior provides additional information to that available from the likelihood alone.

In our case, we use the latter interpretation where we do not acquire our data in sequence but have a good idea about the general location and range of the model parameters. This information comes from the analysis of the problem and of the likely parameter values. We shall examine this more closely in the following chapters.

1.6 The contributions made in this thesis

The main contribution made in this thesis is that encapsulated in the main hypothesis - namely the extension of the linear combination of views theory with appropriate probabilistic constraints and the combination with the resulting MAP estimation with an optimisation algorithm so that it may be used

for solving the recognition problem for 3-dimensional objects using only pixel information and models derived from a small number of nearby 2-dimensional intensity images of the objects of interest.

By initially examining the 2-dimensional image-based object recognition problem in detail, we soon realised that efficient and accurate recovery of the optimal transformation parameters that would bring a model and a scene object into agreement required the use of probabilistic constraints in the transformations. Additionally, we discovered that it was essential to consider the transformation T as a product of independent, primitive transformations, each assigned a separate prior distribution. Such a separation of the degrees a freedom revealed that the primitive transformations are not equally likely in a typical object recognition setting and should be biased differently. The use of such priors in a Bayesian model together with the use of a powerful optimisation algorithm produced very good recognition results without the requirement for extensive off-line training, time consuming search or the need for good initialisation. The same principle was then extended to multiple views in 3-D and to the LCV paradigm.

As a result, we developed a system that can recognise 2-dimensional intensity projections of 3-D objects from a variety of poses via a small number of stored views of each of the objects of interest. The system may be applied to a variety of elaborate problems in different recognition scenarios and is very simple to set-up (generate a database of models) and use (no need for good initialisation or complicated configuration of the optimisation algorithm).

The work carried out for this thesis has also produced a number of secondary novel ideas and results, the most interesting of which we list here:

- **Analysis of the posterior space both graphically and numerically**: During the course of our research we explored the properties of the error space near the optimal solution, collecting both graphical and numerical information. This gave us valuable insight into the complexity of the space under various recognition set-ups (e.g. simplistic versus more elaborate backgrounds) which in turn allowed us to adjust our model and solution approaches accordingly. Information on error surfaces not previously seen in such detail is introduced in this thesis.

- **Comparison of different error metrics**: In our attempt to discover a good error metric well suited to the specific needs of image-based template matching we compared different solutions, such as use of: the normalised cross-correlation, the Huber norm and mutual information, each of which produced different error surfaces and as a result, different optimisation results. This information can now be exploited in other applications where pixel intensity is used and the solution depends on the scene complexity, the type of object of interest and the imaging process.

- **Comparison of different optimisation methods**: For recovery of the optimal model transformation parameters it is essential to choose an appropriate optimisation method. That generally means an algorithm that enables one to find a good-enough solution as early as possible in the computation, without the need for time-consuming parameter tuning or strict initialisation. Furthermore, the algorithm should, in general, improve quickly on discovery of a good solution. As a consequence, we contrasted several solutions in a number of problems with varying degrees of difficulty. In addition, certain algorithms that we explored such as differential evolution

[Storn and Price (1997)] and SOMA [Zelinka (2004)] have not received adequate attention in computer vision tasks. We believe that the results from this thesis may be relevant in other research involving optimisation on intensity images, such as medical image registration.

- **Extended simplex algorithm**: As part of our investigation into various optimisation algorithms, we used the simplex method developed by Nelder and Mead (1965)] as a way of improving on the discovery of good solutions found by use of other algorithms. The simplex method is a direct search, local optimisation method able quickly to minimise an energy function, but it can easily get stuck in local minima and not make significant progress after the first few iterations.

  We thus extended the basic form of the algorithm by incorporating a *restart step* that allows the simplex to "jump-out" of a local minimum and continue from a nearby location. Furthermore, as the algorithm progresses the jumps get smaller according to an 'annealing' schedule. This modification allows the simplex to burrow further into the error surface, dramatically improving the optimisation results even on functions with multiple local minima. In fact, it may be used as a way of quickly improving the results already identified by slower-converging, global stochastic optimisation algorithms in a hybrid minimisation scheme.

- **Foreground - background model**: In this work we mainly focused our efforts on building robust geometrical models for the objects in the foreground. This worked well enough, provided that the scene contained trivial (simplistic) background data and there was no change of illumination between the model and the imaged object.

  This however, limited the applicability of our method to synthetic or highly-controlled scenes, or where the background was explicitly provided as a separate entity. Near the end of our research we experimented with inclusion of a background model, first in the 2-D approach and later in the LCV 3-D approach, and incorporated a basic affine model to accommodate illumination changes. Although developed theoretically, we did not have the time systematically to test these new models in extensive experiments. These models however represent a significant first step in extending the LCV equations correctly to deal with background data and accounting for the additional degrees of freedom from lighting variations.

1.7 The significance of this work

The work we have carried out in this thesis is one of the first systematic attempts to use view-based techniques which allow pose-invariant modelling and recognition of 3-D rigid objects directly from 2-dimensional intensity images using pixel information alone. Neither feature extraction nor the establishment of a dense correspondence is necessary at any time during the model building or recognition stages.

We thus anticipate that the probabilistic LCV method owing to its practicality, ease of initial set-up and use and its good results across a range of different objects will be useful in a variety of applications including, but not limited to:

- robotic and autonomous navigation,
- medical image registration and data extraction,
- object tracking, and
- automated control and access systems.

1.8 Papers resulting from this thesis

In the course of the work described in this thesis, seven papers have been produced for publication at conferences and in journals. They represent various stages in the development of our approach and are listed below in chronological order:

- V. Zografos and B. F. Buxton, *"Affine Invariant, Model-Based Object Recognition Using Robust Metrics and Bayesian Statistics"*, International Conference on Image Analysis and Recognition (ICIAR) **2005**, pp. 407-414.
- B. F. Buxton and V. Zografos, *"Flexible Template and Model Matching Using Intensity"*, Digital Image Computing: Techniques and Applications (DICTA) **2005**, pp. 438-447.
- V. Zografos and B. F. Buxton, *"An evaluation of common distributional models for a Bayesian prior of the scale transformation"*, initial draft prepared for submission to Elsevier Science **2006**.
- V. Zografos and B. F. Buxton, *"Pose-invariant 3-D object recognition using linear combination of 2-D views and evolutionary optimisation"*, International Conference on Computing: Theory and Applications (ICCTA) **2007**, pp. 645-649.
- V. Zografos and B. F. Buxton, *"Evaluation of linear combination of views for object recognition"*, in Advances in Intelligent Information Processing: Tools and Applications, **2007** ed. B. Chanda and C. A. Murthy, World scientific, pp. 85-106.
- V. Zografos and B. F. Buxton, *"A Bayesian approach to 3-D object recognition using linear combination of 2-D views"*, 3^{rd} International Conference on Computer Vision Theory and Applications (VISAPP) **2008**.
- V. Zografos *"Comparison of optimisation algorithms for deformable template matching"*, Submitted to ISVC **2009**.

1.9 Definitions

In this section we include in order to avoid confusion some definitions of a number of terms commonly used in this thesis that may, in publications, have more than one shade of meaning. These are:

- **Corresponding landmark points**: By corresponding landmark points in two or more images we mean landmark points in each image which are projections of the same 3-D world points, marked on the imaged object or scene (i.e. a correspondence in a stereo vision sense).

- **View/pose**: We shall not distinguish between a view of an object and its pose since variations in either cause the same affects in a captured image.
- **Basin of attraction**: This is a region in the solution space of an algorithm in which all starting points converge to the same solution, or possibly cycle of solutions.

1.10 Abbreviations

- **LCV**: Linear Combination of Views
- **DE**: Differential Evolution
- **SOMA**: Self-Organising Migrating Algorithm
- **CATT**: Centred Affine Trifocal Tensor
- **ISPM**: Integrated Shape and Pose Model
- **PCA**: Principal Components Analysis
- **ASM**: Active Shape Model
- **MAP**: Maximum A-Posteriori
- **ML**: Maximum Likelihood
- **NFEs**: Number of Function Evaluations
- **MCMC**: Markov-Chain Monte-Carlo
- **d.o.f.**: Degrees of Freedom
- **pdf**: Probability Density Function
- **cdf**: Cumulative Distribution Function
- **SSD**: Sum of Squared Differenecs
- **SAD**: Sum of Absolute Differences
- **CC**: Cross-correlation
- **BP**: Back-projection

1.11 Structure of this dissertation

The rest of this dissertation is organized as follows. Chapter 2 contains a review of the relevant literature which is intended to locate our work within the context of previous research. Chapter 3 introduces the theoretical background upon which this thesis is based and offers a summary of what are the most important and recent topics in model-based object recognition. In chapters 4,5 and 6 we present the main contribution of this thesis, starting from 2-D object recognition and expanding into 3-D, followed by our

work with optimisation algorithms. Chapter 7 presents the analytical experiments of the probabilistic LCV method on synthetic and real datasets and an exploration of different error measures for intensity-based, template matching. We make use of chapter 8 to provide our thesis conclusions and offer some possible avenues for future research work in this area. The bibliography follows at the end.

Chapter 2

Related work

Object recognition in its general form has been widely studied and a plethora of different approaches exist that attempt to solve different aspects of this problem depending on the application area. These approaches vary according to the type of knowledge they employ, the restrictions placed upon the objects recognised (for example objects may be 2-dimensional or 3-dimensional, simple or complex, rigid or flexible), the object representation and coordinate system used, and the overall strategy employed. In this chapter, we will closely examine the main ideas behind recent research methods in object recognition. In particular, we will consider model-based methods, in which prior knowledge of the object's appearance is provided by an explicit model as these are most relevant to our research.

2.1 Choice of coordinate system

The first step in an object recognition system is to define an appropriate coordinate system. There are two ways to define this coordinate system for a three-dimensional shape, the *viewer-centred approach* and the *object-centred approach*. Since images represent a scene from a camera's perspective, it is only natural to represent objects in a viewer-centred coordinate system. Nevertheless, it is easy to transform from one coordinate system to the other and use an object-centred approach instead. The main reason behind choosing one system over the other is efficiency in representation for feature detection and subsequent low-level processing. A representation allows certain operations to be more efficient at the expense of others, so obviously a choice has to be made based on the requirements of the application at hand.

2.1.1 Viewer-centred approach

If objects usually appear in a relatively few stable positions with respect to the camera then they can be represented efficiently in a viewer-centred, viewing angle dependent, coordinate system, which describes the 3-D object using a set of 2-dimensional characteristic views or aspects. Each characteristic view describes how the object appears from a single viewpoint. Typical examples of object recognition using viewer-centred representations are the aspect graphs by [Koenderink (1990); Poggio and Edelman (1990); Bülthoff and Edelman (1992); Ullman and Basri (1991)].

Matching in such approaches is straightforward because it involves comparing descriptions that are both 2-dimensional. There is no need for model projection during matching and the continuous space of viewpoints has been reduced to a discrete space of characteristic views. If the camera is far away

from the object(s) of interest then, under such *affine imaging conditions* their three-dimensionality can be ignored and objects may be represented sufficiently well by a limited set of views.

The disadvantage of using a viewer-centred representation is that for moderately complex objects, in principle because of the large number of different aspects they may present, a large number of different views need to be stored thereby increasing the storage space requirements relative to object-centred approaches. This also means that, in the object matching stage, many more models need to be considered, since each characteristic view is a separate model. Even so, testing each model is far less computationally expensive than in the object-centred approach, since we are dealing with a 2-D instead of a 3-D match. Furthermore, in practice, many of the aspects of an object differ only in small details and occupy only a small portion of the view-sphere and may, for many object recognition purposes, be ignored.

Viewer-centred representations have become quite popular, as there is some interesting evidence that the human visual system uses a similar representation for object recognition [Bülthoff and Edelman (1992); Tarr et al. (1998); Tarr and Büllthoff (1998)]. Experiments have shown that humans are able to recognise objects accurately and rapidly from particular viewpoints, which implies that those views of the object are readily available (stored in memory) while others are computed as needed. In addition, the availability of large amounts of RAM in modern computers (several GByte at the time of writing) makes such an approach more attractive as it suggests trading computation for memory.

A viewer-centred representation, however, only provides an approximation to the object's shape and appearance. Each characteristic view represents a range of viewpoints over which the object varies in shape and appearance. The more characteristic views we use, the smaller the range each view covers and the more accurately the object is depicted over that range. We therefore have a trade-off between the size of the description and its accuracy. One way to deal with this problem is to take advantage of certain invariant features that exist among a range of viewpoints. For example, certain relations between lines (co-termination, parallelism, co-linearity), angles between lines and ratios of line lengths are invariant with respect to view point. Use of such techniques can extend the range of viewpoints covered by a characteristic view and thus improve the trade-off between accuracy and number of views. Another way is to interpolate between characteristic views. As we will see later on, this can be achieved via the Linear Combination of Views method, where a new view can be constructed from 3 or more stored views and a linear operator.

2.1.2 Object-centred approach

The alternative to the viewer-centred approach is the object-centred approach, which describes objects usually as a three-dimensional entity within a coordinate system attached to the object. [Marr (1982)] for example, specified the object's parts relatively to the object's main axis. Object-centred representations are independent of the camera parameters and location and yield the most concise and usually most accurate shape descriptions. However, in order to make them useful for object recognition, the representations should have enough information to produce object images or object features for a given camera parameterisation and viewpoint. This suggests that an object-centred representation should explicitly capture aspects of an object's geometry. Some common such representations are: *constructive*

solid geometry, where simple geometric primitives are used together with Boolean operators to represent an object and *spatial occupancy*, where an object in 3-D space is represented by using non-overlapping sub-regions of the 3-dimensional space occupied by an object, such as a voxel representation, octree or tetrahedral decomposition.

When object-centred coordinate systems are used for model description in object recognition we must do one of the following: either i) derive a similar object-centred description from the image and try to match that description with various models, or ii) derive a 2-D description from the image, and use a matching procedure combined with a projection of the 3-D object to the same 2-D image description. [Lowe (1985)] does exactly that by projecting each 3-D model stored in memory to a hypothetical viewpoint and matching the resulting projected locations of the 2-D features to the input image. A similar idea is presented by [Ullman (1989)] in his recognition by alignment approach.

2.2 Choice of strategy: features vs templates

There are also two main choices for the object recognition strategy: the *feature-based* strategy, which is based on shape information [Huttenlocher and Ullman (1990); Lamdan et al. (1988); Jacobs (1997)] and the *image-based* strategy, which is based on direct representation of image intensity [Murase and S.Nayar (1995); Turk and Pentland (1991); Borotschnig et al. (2000)] or on a filtered version of the image [Sullivan et al. (2001); Srivastava et al. (2002)].

2.2.1 Feature-based approach

This computational strategy for object recognition is based on the idea that much of the information about an object is encapsulated by its geometrical properties. It usually relies on a geometrical model of an object's shape characteristics which is often applied to simple data, and is used to explore the correspondences between the model's features and the detected features in the scene during recognition.

Given an unknown scene and an object model, both represented in terms of their features, in this approach the objective is to find a partial match between the two and estimate the object's location and pose in the image. A match solution must satisfy the viewpoint consistency constraint [Lowe (1987)] which stipulates that the locations of the object's features in the image must be consistent with some pose of the object. We are essentially looking for the transformation T that will bring the two corresponding sets into alignment. These sets of features are usually stored in n-dimensional vectors, and matching is carried out by minimising some dissimilarity metric, or measure of quality, over the parameters of the transformation T. Such measures of matching quality are often based on error models that describe how image features differ from model features. Two common error models are: i) a *bounded error model* which requires that each image feature is positioned within some fixed range of its predicted location. The related match quality measure is usually just the count of matching feature; and ii) a *Gaussian error model* which assumes that image features are distributed normally and independently about their predicted locations. The related match quality usually considers both the number of matching features and the sum of squares of their normalised errors.

Since usually there is no a-priori information as to which model features (or parts) correspond to

which scene features (or parts) it is necessary to solve the correspondence problem. If we consider, for example two sets X and Y, each containing N points, we need to ensure that each point x_i in the image corresponds to the same physical point y_i on the object or projected from the object. Only then are the two sets in correspondence. This requirement makes feature-based recognition computationally expensive even for a moderate number of features, especially if feature detection in the image is imperfect and there are false positives (false alarms due to clutter or other objects) and false negatives (features missing dies to lack of sensitivity). Traditional object recognition systems thus often lack scalability especially when faced with a large number of models, when image features cannot reliably be grouped object by object, or extensive variations in object appearance are encountered [Binford and Levitt (1996)]. To limit the possible number of matches, methods have been proposed based on geometric constraints such as the interpretation tree by [Grimson and Lozano-Perez (1986)], or minimum number of feature correspondences [Huttenlocher and Ullman (1990)] and early localisation [Faugeras and Hebert (1983)]. The method of indexing [Califano and Mohan (1994)] is an alternative approach that uses a-priori information quickly to eliminate inappropriate matches during recognition.

Some common paradigms of feature-based object recognition that deal with changing object geometry due to pose variations include the use of invariants [Leung et al. (1998); Maybank (1998)], explicit 3-D models [Blanz et al. (1996); Lee and Ranganath (2003)] and multiple views [Lamdan et al. (1988); Binford and Levitt (1996)]. The first paradigm makes use during recognition of special invariant properties of geometric features (i.e. properties that vary little or not at all as viewing conditions change). The most serious problem with this method is that quite often it is very difficult, if not impossible, to find general geometric invariants. For example, no such invariants exist for single images of 3-dimensional objects under a 3-D perspective projection [Clemens and Jacobs (1991)]. The second paradigm employs a full, explicit 3-D model to which the image formation process is applied during recognition. This is in fact a projection operation that generates new images of the object that can be compared with a given scene. This idea works well if we have a 3-D model of the object - which is not always practical - and provided that we know the specifics of the image formation process - which may not always be the case. In the last paradigm, an object is modelled by a set of 2-D reference views that describe how the shape of the object varies across different views on the viewing sphere. Such methods perform recognition by matching the novel view with one of the reference views, or at least a part of it. This strategy is quite inefficient since a large number of views must be stored for each model, unless we utilise some of the techniques we have seen in Section 2.1.1. Range and colour have also been employed in applications such as face detection [Kim et al. (1998)]. In this work disparity maps are computed and objects are segmented from the background by means of a disparity histogram. They use a Gaussian distribution in normalised RGB colour space that classifies segmented regions with skin-like colour as faces. A similar approach has been proposed by [Darell et al. (2000)] for face detection and tracking.

In addition, one can classify the various feature matching methods according to whether they search for a solution in correspondence space, transformation space or a mixture of both. Correspondence space is the space where sets of image and model features are paired together. Transformation space is the

space of possible transformations between the object and the camera. Under the viewpoint consistency constraint and with an appropriate error model the two spaces are closely related, with each match being consistent with a set of transformations and each transformation with a set of matches. Typical exemplars of correspondence space search are the interpretation tree [Grimson and Lozano-Perez (1986)] we have mentioned above, and graph matching methods [Siddiqi et al. (1999); Caelli and Kosinov (2004); Marcini et al. (2002); Wiskott et al. (1999); Bergevin and Levine (1993)] where one tries to find a partial match (sub-graph isomorphism) between a graph that represents the model's features and a graph that represents the detected image features. The biggest problem with correspondence space methods is their computational cost which is generally exponential in the number of model features. Some techniques such as relaxation (see [Grimson (1990)] for the heuristic search termination method) whereby we settle for a near-optimal match can help alleviate the computational problem. When it comes to transformation space search methods, the generalised Hough transform [Ballard (1981); Grimson and Huttenlocher (1988)] is one such representative example. Methods that search the transformation space generally avoid the costly exponential search. Alternatively, we could also use a mixture of the two methods, and carry out a portion of the search in each space. For example, the alignment method of [Ullman (1989)] begins the search in correspondence space until it matches enough reference features to determine the viewpoint transformation.

There are of course many questions that need to be addressed when using a feature-based approach. For example, what kind of features should we detect and how can we detect them reliably and efficiently? Most features can be computed in 2-D images, but they are related to 3-D characteristics of the objects. Owing to the nature of the image formation process some features are relatively easy to compute while others can be very difficult. We also need to establish how features in images can be matched to models stored in a database. In most object recognition tasks, where there are many features and numerous objects, methods such as exhaustive searching may solve the problem but are probably too computationally costly to be useful. Effectiveness of features and efficiency of a matching technique must be considered when choosing an object recognition strategy.

2.2.2 Image-based approach

A desirable characteristic of image-based recognition is that object models can be compared directly or fairly directly with input data, as both are of the same type (e.g. intensity images). Feature-based methods instead require that features be detected and described before data and model can be compared. This means that in distinction to the procedure in feature-based approaches, an image-based approach does not need to recover the geometry of the objects but can learn their appearance characteristics from training imagery. A model of the object is built off-line from a collection of different images depicting a variety of object appearances taken under changing viewpoints and lighting conditions. In this way, each model view is stored as a vector of image intensities in some low-dimensional space that captures the significant characteristics of the object, such as the eigenspace [Murase and S.Nayar (1995); Lamdan et al. (1988)]. A hyper-surface in this space represents a particular object. Recognition is carried out by projecting the image of an object to a point in the low-dimensional space. The object is recognised by calculating

the shortest distance from a given hyper-surface. The location of the point determines the pose of the object. Other image intensity methods include use of colour histograms [Vinod and Murase (1996)] and photometric invariants [Schmid and Mohr (1997)]. More recently, [Gross et al. (2004)] have used the light-field of an object as a set of features projected into a low-dimensional eigenspace. This way they have captured radiance values from arbitrary illumination conditions and with the use of a classification algorithm have applied this theory to face recognition across a range of poses.

There are of course simpler ways for fitting intensity models directly to photometric data. We can divide such methods into *rigid model fitting* and *flexible model fitting*. In rigid methods, the shape or photometry of the target object is known beforehand in the form of a template. The template may represent an object as a rigid curve or an image and is matched to the image data by means of a metric that may represent either a similarity or dissimilarity measure. Where that metric is (say) maximal, we have the optimum template location and therefore a match. The simplest such metric is normalised cross-correlation, which has been applied successfully in grey-scale and colour imagery with the use of an exhaustive search technique [Tsai et al. (2003); Tsai and Lin (2003)]. Rigid model techniques are ideal when the object shape or photometry are precisely specified because of their restricted search space. In addition, they are relatively insensitive to noise. Nevertheless, when the exact object shape or photometry is not known, or when we have to deal with many model types at the same time, or even in the case when they have to be applied over foreground and background without an explicit background model, such methods should generally be avoided. It is possible, however, to consider variants of the technique , such as geometric hashing [Lamdan and Wolfson (1988); Grimson and Huttenlocher (1990)], in order to deal with fitting a large database of models simultaneously.

In the case where the above application of a rigid template is not possible, flexible model fitting techniques may be more useful. These methods support the use of models that are governed by a number of generic constraints on object characteristics (e.g. smoothness, curvature, compactness, symmetry and homogeneity) and rely on an optimisation procedure that finds the best fit between the model and the image data. The fit of the model to the image is measured by an objective function and matching is performed by (say) minimising this measure. Like template matching, flexible model fitting operates at the pixel level but because of the additional degrees of freedom that the flexibility allows, the search may become computationally expensive. Therefore, flexible methods normally require a good initialisation close to the basin of attraction of the correct match or the use of heuristics to control the search and reduce the computation at the possible risk of a non-optimal solution. As noted in the introductory chapter, the basin of attraction is the region of the solution space within which an iterative optimisation method will converge to an optimal solution, or solutions.

The most severe limitation of the intensity-based approach is that it requires isolating the object of interest from the background. This approach has been demonstrated successfully on isolated objects or pre-segmented images, but has been difficult to extend to cluttered and partially occluded scenes. There have been a number of attempts to improve robustness to occlusion, such as using small eigen-windows [Ohba and Ikeuchi (1997)] and parts from objects [Huang et al. (1997)] but such methods have extensive

search requirements or rely on explicit 3-D models.

Image based methods can thus be successful in handling the combined effects of shape, pose, reflection and illumination, but have serious difficulties in segmenting the object(s) from the scene and dealing with occlusions. Since matching is performed directly in the image domain, rather than in the geometric feature domain, performance is not affected by increasing geometric complexity. A great advantage of image-based methods is that any shape can be represented no matter how complex as long as we can take images of it. Relevant work by [Brunelli and Poggio (1993)] on comparing the two approaches in their simplest form, has shown that template matching is superior in object recognition performance and simpler in use. The feature-based strategy, however, may allow a higher recognition speed and smaller memory requirements.

2.3 Choice of model representation

Object recognition techniques can also be categorised according to their choice of model representation. These categories have traditionally been: *feature points*, *curves*, *orthogonal basis* and *image templates*.

2.3.1 Feature points

Perhaps the most simple model representation is based on a set of landmark points. These points are chosen in specific locations so as to convey the characteristic shape of an object. For example, along edges and corners of the object boundary, or around important features, such as the eyes, the nose and the mouth in a facial image. Object matching and recognition require that we detect and label similar feature points in the image and match them with our model using some kind of metric such as the sum of squared differences or their coordinates. Feature detection however is very problematic since there is no easy way effectively to detect all the correct features. Feature detection algorithms will often either detect more features than exist in an image, or will not detect all the correct features. Therefore techniques based on sub-graph matching [Caelli and Kosinov (2004)], methods such as the interpretation tree [Grimson and Lozano-Perez (1986)], are necessary to overcome the feature detection problem.

Some of the existing challenges for object recognition (as we shall see in detail in later chapters) is missing or corrupt data possibly due to occlusion, disjoint training and testing sets and the existence of noise. Recent feature based methods such as the recognition-by-parts approaches, originating from the early attempt by [Biederman (1993)] to model pattern recognition in terms of how a human observer learns to discern patterns from their constituent parts, have recently been re-visited by the research community [Stommel and Kuhnert (2009); Vasanthanayaki and Annadurai (2005); Amit and Trouve (2007)] in order to overcome these problems. In part-based models, a small number of features and their relations (for example relative distance [Fergus et al. (2003)]) are used in order to determine if an object is present in the scene and therefore they can deal, to some extent, with incomplete data.

2.3.2 Curves

One of the first and most popular curve-based representations is a labelled set of points with connectivity information. This representation is similar to a vertex and edge representation (e.g. a polygon or a linear spline). Numerous authors have used this point set representation, such as [Burr (1981);

Jolly et al. (1996)]. Another popular representation is B-splines, which uses continuous curves to model the geometry of an object. Compared to the point set representation, B-splines have the advantage of a lower dimensional parameter space since a B-spline can be obtained via a few control points. In addition, B-splines have the additional, advantageous property of inherent smoothness. [Cipolla and Blake (1990)] and [Menet et al. (1990)] were the first to develop deformable models using B-splines after which they have been used in a number of different studies [Blake and Isard (1998); Isard and Blake (1998); Klein et al. (1997)]. Finally, another representation that has received much attention in the literature is the use of level sets [Sethian (1999)]. Compared to other methods, level sets have the advantage of allowing automatic merging and splitting of the initial contour. Research on level sets for object recognition that may be regarded as characteristic of the field is that of [Paragios and Deriche (2000)] and [Leventon et al. (2000)].

2.3.3 Orthogonal basis

Orthogonal basis representations usually apply a reduced or truncated parameter space in which only the most important characteristics and descriptors are used. Perhaps the most widely used such representation is the Point Distribution Model, proposed by [Cootes et al. (1995)]. According to their method an object is represented by the mean shape of a training set and a linear combination of the most important eigenmodes of the shape variation from this mean. The Point Distribution Model has played an important role in the popular Active Shape Model [Cootes et al. (1995)] and has been extended with texture in the Active Appearance Model [Cootes et al. (2001)]. Numerous other models such as that of [Duta and Sonka (1998); Dias and Buxton (2002)] have been based on this representation. Other representations are Fourier descriptors [Staib and Duncan (1992)] which use trigonometric functions as the orthogonal basis and Wavelet descriptors [Yoshida et al. (1997)] that are defined as dilated and translated versions of a basis wavelet. A comparison of shape models based on the above mentioned orthogonal basis representations can be found in [Neumann and Lorenz (1998)].

2.3.4 Image templates

The last representation we will examine here is that of a prototype image template. Such a representation is used in object recognition, and may be deformed under a similarity or affine group of transformations to match a new object in an image. Most of these models can be classified as registration methods. Typically, the template is the same type as the image (i.e. intensity data) but edge templates have also been used [Jain et al. (1996)]. There is a rich collection of examples of this representation and some of the best known are [Amit et al. (1991)], [Christensen et al. (1996)] and [Sclaroff and Isidoro (1998)].

Even though it is difficult to answer the question as to which of the considered representations is the best, there are a number of properties which, though from a general point of view they are desirable, by no means make a certain representation superior. These are:*generality*: the representation should be able to model an arbitrary object. *Specificity* : the representation should enable particular objects, or object classes, to be distinguished from others. *Low dimensionality* : a low dimensional representation with little redundancy improves the computational efficiency and makes optimisation easier and more robust. And finally, *linear parameterisation*: a restriction to linear parameterisation has certain advantages in

simplifying fitting algorithms and avoiding problems with local minima [Blake and Isard (1998)].

2.4 Deformable template models

We have chosen to present deformable template models here separately since they not only constitute the main focus of this work but also comprise a substantial proportion of recent research into object recognition. Perhaps the most thorough review on deformable template models is that by [Jain et al. (1998)] from which we have adopted the classification of different deformable template methods.

If we were to start with a definition, we could say that deformable template models are models which under an implicit or explicit optimisation criterion deform to match a known type of object in a given image. Alternatively, we could recall that deformable models were designed to overcome one of the most important obstacles to object recognition; that is, the integration and interpretation of different local image cues (intensity, gradient, texture etc). In addition, of course, they also overcome the fact that exact geometrical models of objects may not always be available because of the variability in the imaging process and inherent within-class object variability. On the one hand, traditional approaches like those we have seen in this chapter cannot cope with adverse imaging and viewing conditions, occlusion and noise. On the other hand, model-free representations fail to converge to reasonable solutions owing to the highly unconstrained nature of the problem. Deformable model matching, is a more powerful technique because of its capability to deal with shape and appearance variations, or as [Jain et al. (1998)] put it:

> *"...deformable models, which have been receiving increased attention lately, provide a promising and powerful approach for solving computer vision problems, because of their versatility and flexibility in object modelling and representation."*

A deformable model is able to adapt to fit the given data and in that sense it can be considered *active*. It is a useful representation, because of its ability both to impose constraints (geometric or photometric) on the model but also to integrate local image evidence. Different deformable template approaches that have appeared in recent literature can generally be partitioned into two main classes: *free-form* models which can represent any arbitrary shape as long as some general constraints are satisfied, and *parametric deformable* models that are able to encode a specific characteristic shape and its variations. This shape can be characterised by a parametric formula or by using a prototype shape and some deformation scheme. Fig. 2.1 illustrates this classification.

2.4.1 Free-form models

Free-form models have no global template structure and apart from some general regularisation constraints, such as continuity and/or smoothness, they can be deformed to match any salient image feature, using, for example, potential energy fields produced by these features. One of the most widely known free-form models is the active contour (snake) popularised by [Kass et al. (1988)]. In this approach, an energy minimisation contour is controlled by a combination of physics-inspired forces or energies that impose constraints on how its shape may very over space and time. This physical interpretation considers models as elastic bodies that respond naturally to externally applied forces and elasticity constraints.

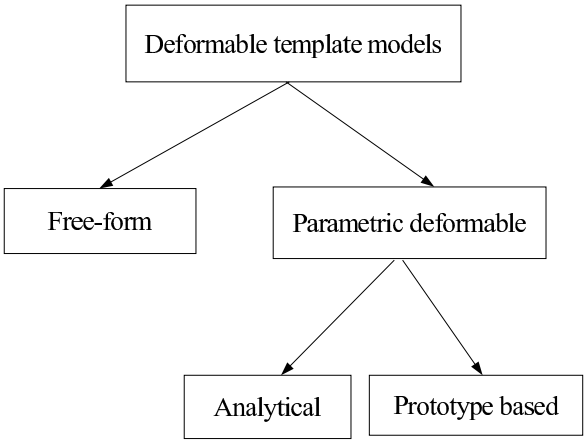


Figure 2.1: The classification of deformable template models in recent research.)

More specifically an active contour is governed by: an *internal contour* energy E_{int} which may enforce smoothness, an *image potential* energy E_{img} that attracts the snake to significant image features, and an *external potential* energy E_{ext} which deforms the model. Each force creates its own potential field and the contour actively adjusts its position and shape until it reaches a minimum of the total snake energy:

$$E_{snake} = \int_0^1 \{E_{int}(v(s)) + E_{img}(v(s)) + E_{ext}(v(s))\}\, ds \tag{2.1}$$

where s is the parameterisation of the contour and $v(s)$ is a point on that contour. Given an appropriate initialisation the snake can quickly converge to a nearby energy minimum. However, the active snake model is essentially a "myopic" approach since it uses only local information and it is very vulnerable to image noise and sensitive to choice of its starting position. To overcome these limitations, researchers have experimented with different energy forces, such as attractors and tangent constraints [Fua and Brechbuhler (1996)], gradient vector flow [Xu and Prince (1998)] or different optimisation algorithms [Cox et al. (1996)].

A similar approach to snakes is that of spline-based deformable models [Figueiredo et al. (1997)], which though they do not encode specific shape information, have more structure than snakes since they are expressed as a linear combination of a set of basis functions. Their shape is defined by the coefficients of these basis functions. However, because selection of coefficients can be arbitrary spline-based deformable models cannot represent a "default" shape when prior information is presented. For that reason, spline-based methods under-perform compared to more appropriate strategies such as the parametric deformable models we will discuss below.

2.4.2 Parametric deformable models

Parametric deformable models are commonly used when prior information about the shape or appearance of the object is available. A characteristic model derived from a set of training images and its variations is encoded using a small number of parameters, achieving thus a compact representation of the object's shape and photometry. There are two general ways to carry out the parameterisation, an *analytical* or a *prototype-based* parameterisation.

Analytical

With an analytical parameterisation, one can represent the geometric shape of an object using a set of analytical curves (e.g. ellipses) and a number of parameters that uniquely describe the chosen set. The shape of the template can be changed by using different values for the parameters and variations in shape are determined by the distribution of the admissible parameter values. Common techniques based on analytical models are the example by [Yuille et al. (1992)] in which they designed parametric models for eye and mouth templates using circles and parabolic curves in order to extract facial features. Also [Lakshmanan et al. (1995)] have used a parametric template to locate the airport runway boundary in radar images. Based on prior knowledge, they derived a global shape of the runway parameterised by the slopes and intercepts between the edges of two parallel lines. Finally in [Jolly et al. (1996)], polygonal templates are used to characterise a general model for a vehicle and to segment vehicles from outdoor traffic scenes.

All these techniques require a good initialisation of the model in order to obtain correct solutions and in addition the approximate pose of the object to be recognised is assumed known. Analytical deformable models have limited applicability because the objects under investigation must have a well-defined shape that it is possible to represent by a set of curves and with a few parameters.

Prototype based

Prototype deformable models on the other hand are more flexible since they are derived from a set of example images. Grenander with his pattern theory [Grenander (1993)] was the first to present a systematic framework for representing a general pattern from a class of shapes. A shape is represented by a set of parameters, different values of which give rise to different shapes. A probability distribution on the parameters is also specified that allows for a flexible bias toward a particular shape. [Grenander and Keenan (1993)] formulated a global, pattern-theoretic model of shape which provides a structured method to generate pattern from a class of shapes. This model can be represented by: i) a prototype template which describes the overall appearance of the shape and ii) a parametric statistical mapping that controls the random variations in the shape class. The prototype template is usually chosen based on prior knowledge of the objects of interest and the parametric statistical mapping is chosen to reflect the allowed deformations on the prototype template.

The success of these models depends on how well the parameters and the probability distributions can be defined accurately to represent the shape and its variability. Indeed, many researchers have used a variety of choices for the prototype template and its possible deformations. For example, [Grenander et al. (1991)] have used polygons to approximate the contours of human hands while variations were described using a Markov process on the edges. [Jain et al. (1996)] used a grey-scale bitmap of the mean object shape with edge information as a way to represent the prototype template. They used parametric transformations with normally distributed parameters to deform this prototype bitmap to match the image. [Zografos and Buxton (2005a)] have expanded on this by working directly with pixel values, introducing more suitable prior distributions and treating the residuals with a robust error norm. [Cootes et al. (1995, 2000, 2001)] have proposed the active shape and appearance models where

the object's shape and appearance is learnt from a set of example images. Once the images are aligned and properly annotated, principal components analysis is used to generate an average shape (prototype) along with modes of variation. [Dias and Buxton (2002); Dias (2004)] went a step further and proposed the Integrated Shape and Pose Model (ISPM). The ISPM is an image based-model that is capable of representing images of 3-dimensional, non-rigid objects without confounding the intrinsic shape variations of the object with the extrinsic pose variations. The ISPM has been shown to outperform Cootes et al.'s Flexible Shape Models and to be a more viable approach than the coupled-view Flexible Shape Model [Cootes et al. (2000)]. Recently, [Felzenswalb (2005)] proposed a deformable model that represents shapes as unique triangulated polygons using constrained Delaunay triangulation. He uses an energy function conditioned on each triangle that has a data term, which attracts the template to the image, and a penalty term that penalises large deformations. The match is located at the point where the transformation has the lowest cost and is found by using a non-serial dynamic programming method that obviates the need for a good initialisation. His technique is not good for objects that may have approximately the same global shape, but have differences in their interior (e.g. faces where their boundary is pretty much the same, but internally they have different features). Such intricacies cannot be captured efficiently by Felzenswalb's method.

2.5 Support vector machines

Recently, methods such as Support Vector Machines (SVMs) have become quite popular in object recognition and thus we mention them here for completeness. SVMs can be considered as a new paradigm to train polynomial function, neural networks or radial basis function (RBF) classifiers. While most methods for training a classifier are based on minimising the training error (i.e. empirical risk), SVMs operate on another induction principle, called structural risk minimisation, which aims to minimise an upper bound on the expected generalisation error. An SVM classifier is a linear classifier where the separating hyperplane is chosen to minimise the expected classification error of unseen test patterns. This optimal hyperplane is defined by a weighted combination of a small subset of the training vectors, called the support vectors. Estimating the optimal hyperplane is equivalent to solving a linearly constrained quadratic programming problem. However, the computation in both time and memory can be intensive. Typical examples are by [Osuna et al. (1997)] where they developed an efficient method to train an SVM for large scale problems and applied it to face detection. Also [Papageorgiou and Poggio (2000)] have used an SVM system to detect faces of pedestrians in the wavelet domain. [Li et al. (2000, 2004)] have used a support vector for determining the pose of an image by using it to choose among face detectors arranged on the viewing-sphere. Face detection is carried out by a combination of Eigenfaces and SVM methods. [Ng and Gong (1999)] achieved real-time, multi-view detection and pose estimation of human faces that undergo non-linear change across the view-sphere. [Pontil and Verri (1998)] used SVMs for 3-D object recognition without the need for feature extraction or pose estimation. An efficient algorithm for training SVMs has been proposed by [Dong et al. (2005)].

2.6 Optimisation

In this section we examine in more detail the use of optimisation algorithms in deformable template matching/registration problems. This will enable us to choose the appropriate solution from amongst a selection of different optimisation strategies to use with our linear combination of views object recognition method. We start with the examination of simple, direct-search methods and move on to more complicated evolutionary algorithms.

2.6.1 Local methods

The tasks of computer vision such as object recognition [Peters (2000)], template matching [Jain et al. (1998)], registration [Brown (1992); Hill et al. (2001)], tracking [Yilmaz et al. (2006)] and classification [Zhou and Aggarwal (2001); Hasegawa and Kanade (2005)] usually involve a very important optimisation stage where we seek to optimise some objective function corresponding to matching between model and image features or bringing two images into agreement. This optimisation stage requires a good algorithm that is able to find the optimum value within some time limit (often in real-time) and sufficiently close to the global optimum.

Traditionally, such tasks have been tackled using local deterministic algorithms[1] such as the simplex method [Nelder and Mead (1965)], Gauss-Newton [Nocedal and Wright (1999)] or its extension by [Levenberg (1944); Marquardt (1963)] and other derivative-based methods [Nocedal and Wright (1999)] (see Fig. 2.2). Such algorithms although they usually converge relatively fast need to be initialised near the proximity of the global optimum otherwise they may get stuck inside local optima and converge far away from the correct solution. One way to overcome this problem is to use multi-resolution search techniques [Maes et al. (1999)]. Such techniques, however, often introduce additional challenges like the tracking of optimal points between different resolution levels that slow the overall process and make it prone to errors. In this work we only examine the simplex and the pattern search methods owing to their simplicity, ubiquity and tractability.

Downhill simplex

The simplex method[2] is a self-contained strategy for optimising an objective function in N-dimensional space and, unlike many other methods it does not make explicit use of a one-dimensional optimisation algorithm as part of its computational strategy. The simplex method requires only function evaluations but not their derivatives and although it might not be the most efficient method available in terms of the number of function evaluations necessary, it is a very good solution when we need something working quickly for a problem with a small computational cost.

A simplex is a polytope of $N+1$ vertices in N dimensions, so in 1-D it is a line, in 2-D a triangle, in 3-D a tetrahedron and so on. The simplex is allowed to take a series of steps (see Fig. 2.3) most notably the *reflection* R, where the vertex with the worst function value is projected through the opposite face of

[1] Algorithms that when given a particular input will always produce the same output for a problem that is fully specified and dependent on known quantities.

[2] Also known as the downhill simplex method or the Nelder and Mead algorithm. It is not to be confused with the simplex algorithm [Dantzig (1963)] for the solution of the linear programming problem.

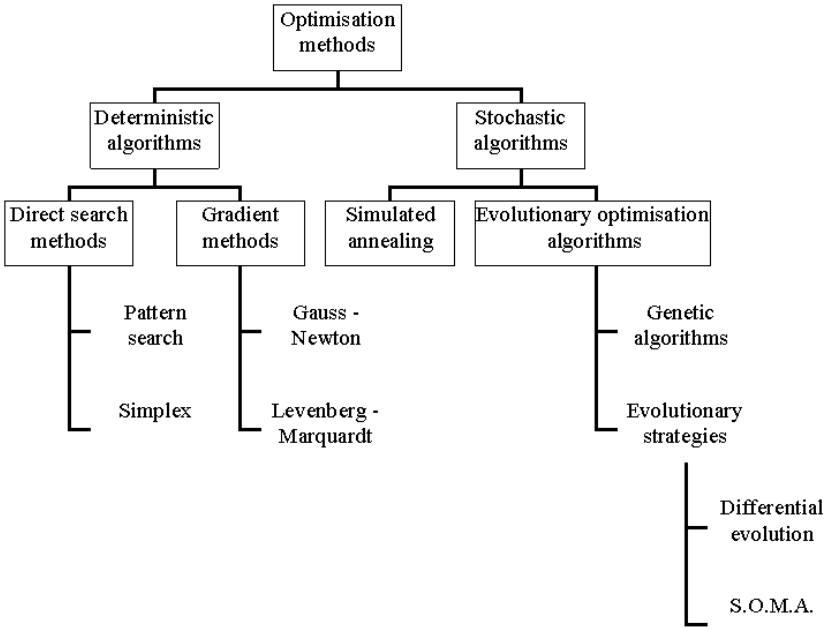


Figure 2.2: Common optimisation methods traditionally used in computer vision.

the simplex to a hopefully better point. The simplex can also change its shape (*expand* E and *contract* C^- and C^+) to take larger steps when inside a valley or a flat region or squeeze through narrow regions. It can also change direction (rotate) by discarding the worst point W when no more improvements can be made and considering the next-worst point amongst the simplex vertices. The simplex must be started not with a single point but with $N + 1$ points so in terms of computational cost, starts and restarts (as we shall see later on) can be expensive. This method is not recommended for problems with objective functions that are costly to evaluate.

We introduced two small yet significant modifications to the basic algorithm [Nelder and Mead (1965)] in order to deal with local optima. The first was the ability for the simplex to *restart* whenever its progress stalled (most likely inside a local optimum). The restart is quite simple. After a number of function evaluations where there has been no change in the value of the tracked optimum we keep the best vertex P_0 and generate N new vertices P_i using the formula:

$$P_i = P_0 + \lambda v_i, \tag{2.2}$$

where v_i are N random[3] unit vectors, $i = 1, \ldots, N$ and λ is a constant which represents the step size. The idea is that by restarting the simplex close to the best point P_0 we can jump out of a local optimum but without jumping so far away from the last good solution we have found.

We also introduced an additional modification which is a reduction of the step size λ from (2.2) based on the number of function evaluations. The rationale behind this is that by reducing λ the overall area of the new simplex is also reduced as the optimisation progresses and it can "burrow" further into smaller areas of the objective surface. This is illustrated in Fig. 2.4. Here we can see all the simplex

[3]This random component will undoubtedly change this particular simplex implementation from a deterministic to a stochastic approach but despite that it still remains a local method.

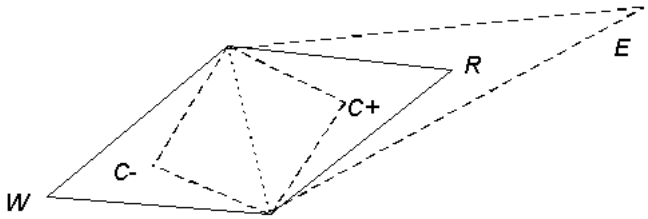


Figure 2.3: Possible simplex moves after the worst point (W) is identified and rejected.

function evaluations (not only the best ones) as it searches the objective surface. In Fig. 2.4(a) we see the simplex using a fixed step size. After a certain point (e.g. ≈200 function evaluations or FEs for short) it stalls and initiates the *restart* procedure. However, the fixed step at that location is too large and the simplex keeps jumping in and out of the discovered good optimum without making any significant improvement for the remaining 800 FEs. Observe now the same experiment with a reducing-step simplex (Fig. 2.4(b)). When this algorithm first stalls it performs big jumps to become unstuck and while it progresses the jumps get smaller as it tries to penetrate deeper into the landscape. If we compare the two methods we can see that in the latter case the algorithm still introduces small improvements driven by the reducing step whereas the fixed step version has stalled many FEs earlier.

We experimented with two reduction schedules, typically encountered in Simulated Annealing [Betke and Makris (1995); Press et al. (1993)]. These are:

$$\lambda = \lambda_0 R^{(k-1)} \tag{2.3}$$

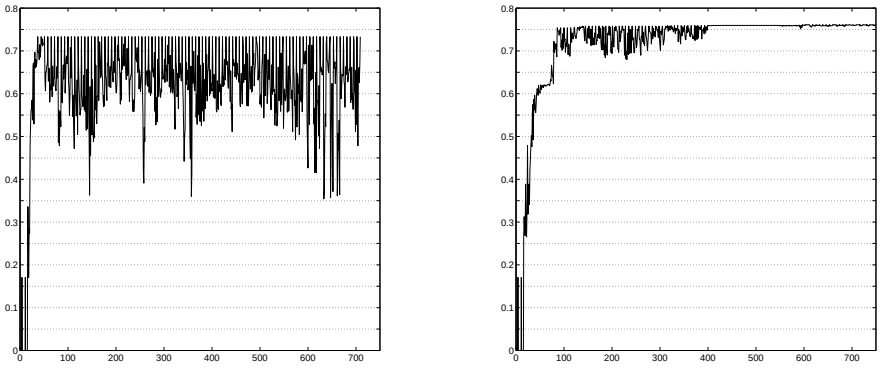
and

$$\lambda = \lambda_0 k^{-1}, \tag{2.4}$$

which are illustrated in Fig. 2.5, with k being the current function evaluation and R the "cooling rate". After some initial tests we decided to use schedule (2.3) since it is more adjustable and changes less abruptly in proportion to any modifications of its parameters. It also does not drop as sharply as (2.4) which means that there is still some significant step length available for later function evaluations. A pseudo-code algorithm of the reducing step restarting simplex is presented in Algorithm 1 in the appendices.

Pattern search

Pattern search algorithms [Audet and Jr. (2003)] are a subset of direct search methods used for solving nonlinear, unconstrained optimisation problems. Similarly to the simplex algorithm, pattern search approaches are considered direct since they neither compute nor approximate the derivatives of the objective function. Direct search methods, as opposed to more traditional approaches that rely on information about the gradient and higher order derivatives to search for an optimal solution, examine the neighbourhood around the current point, looking for a solution where the value of the objective function is lower



(a) (b)

Figure 2.4: Comparison between a fixed (a) and a reducing-step (b) restarting simplex.

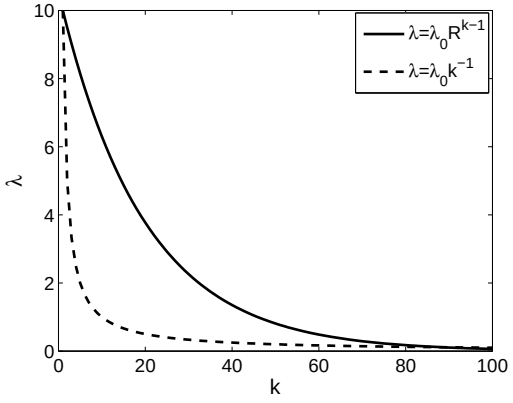


Figure 2.5: The two examined step reduction schedules.

than the current one. As a result, direct methods may be used to optimise objective functions that are non-differentiable or even non-continuous.

A pattern search method proceeds by conducting a series of exploratory moves around the current point x_k, sampling the objective function f in search of a new point (*trial point*) $x_{k+1} = x_k + s_k$ with a lower function value $f(x_{k+1}) < f(x_k)$ (or higher if we are using a similarity metric), where k is the iteration number and s_k is a vector called a *trial step*. The set of neighbourhood points sampled at every iteration is called a *mesh*, which is formed by adding the current point to a scalar multiple of a fixed set of vectors called the *pattern* P_k and which itself is independent of the objective function f. If the algorithm finds a new point x_{k+1} in the mesh that has a lower function value than the current point x_k, then the new point becomes the current point at the next step of the algorithm.

Individual pattern search methods are distinguished by their specific exploratory moves algorithm and they must all satisfy the following two requirements:

- The direction BC_k of any accepted step s_k is defined by the pattern P_k and its length is determined by the step length parameter Δ_k, where $s_k = \Delta_k BC_k$. B is known as a *basis matrix*, and C_k as the *generating matrix*.

- If a simple decrease on the function value is found amongst any of the trial steps of the current iteration, then the exploratory moves algorithm must produce a step s_k that also gives simple decrease on the function value at the current iteration.

Every different pattern search method needs to have the basis matrix B, the generating matrix C_k, the exploratory moves algorithm to be used to produce the step s_k, and a method for updating C_k and the length parameter Δ_k specified. Even so, we can outline a general pattern search algorithm (Algorithm 2), presented in the appendix section, that all individual methods should adhere to.

2.6.2 Global methods

In recent years a wide selection of global, stochastic optimisation algorithms has been introduced, such as the genetic algorithms (GA) [Goldberg (1989)], mainly for engineering problems. Stochastic algorithms are intended for optimising systems where the functional relationship between the independent input variables x and output y of the system is not known. The effectiveness of these algorithms in global optimisation has ensured their use in computer vision applications. Their main advantage is that they are able to find the optimum value without the need for good initialisation. On the other hand they require considerable parameter adjustment which in some cases is not an intuitive or straightforward process. In addition they tend to be somewhat slower than local, deterministic algorithms since it is necessary to use a higher number of function evaluations.

In this section we will introduce certain global optimisation methods, specifically differential evolution (DE) [Storn and Price (1997)] and SOMA [Zelinka (2004)] that appear to be new to computer vision applications and compare them with a traditional approach, that is a generic Genetic Algorithm [Holland (1992)], to determine if these new methods are better suited for solution of typical computer vision problems. We hope to demonstrate how much more suitable such stochastic, global algorithms

are in overcoming typical problems, usually associated with the local methods already mentioned, so that their use in computer vision can become more widespread.

Genetic algorithm

A genetic algorithm (GA) [Holland (1992)] belongs to a particular class of algorithms based on the principles of evolutionary biology such as: *inheritance*, *mutation*, *selection* and *crossover* in order to find the optimum of an objective function. A GA maintains a collection of possible solutions each of which is generated not only by some random perturbation (mutation) but also by a combination of two random solutions from the collection. Suitable candidates for these mutation and combination are chosen by probabilistic criteria. Almost all GAs, no matter how different they might appear, follow these basic stages: *initialisation*, *selection*, *reproduction* and *termination*. What distinguishes one algorithm from another is the variety of ways we can carry out the requirements of each of these stages. In more detail we have:

1. Initialisation: Every GA starts with a randomly selected population of candidate solutions to our optimisation problem (usually called *individuals*) which may be represented in a variety of ways (binary strings, number strings, characters, number vectors). Usually, there is no prior knowledge about the location of the global minimum apart from the approximate boundaries of the system variables (e.g. in the case of template matching: size of scene image, angle of object rotation, magnitude of object scaling and so on), and thus the initial population is generated in order to cover as much of the search space as possible. One factor that is quite important during the initialisation stage because it determines the performance of a GA is the *diversity* of the initial population. If the average distance between individuals is large then the diversity is high whereas if the average distance is small then the diversity is low. A very low diversity will most probably cause the genetic algorithm to stall or converge inside a local optimum, while a very large diversity will slow the progress of the algorithm because of the increased search space. It is quite possible for a GA to find the correct solution even if the latter was not inside the boundaries of the initial population provided the following populations have sufficient diversity. Additionally, we can adjust the diversity of the population after initialisation by increasing or decreasing the amount of mutation. An increase in mutation brings about an increase in diversity and vice versa. Getting the right amount of diversity is usually a process of trial-and-error.

2. Selection: In every generation, a number of the population individuals are selected to reproduce and create a new generation of solutions. Individuals from the current generation are selected through a probabilistic process using fitness-based criteria. In this way, fitter solutions are typically more likely to be selected but a small proportion of less fit solutions will also be included in the next step of reproduction so as to help maintain a high diversity of the population while preventing premature convergence to sub-optimal solutions.

3. Reproduction: The aim of reproduction is to create a new generation of a population of solutions from the current generation using the operations of *crossover* and *mutation*. Once a pair of “par-

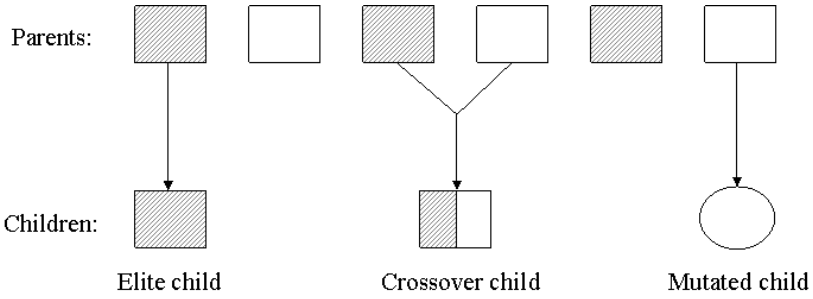


Figure 2.6: Reproduction children in a typical genetic algorithm.

ent" solutions from the current population has been selected, they are genetically combined to produce a "child" solution that retains some of the characteristics of its parents. This process continues until a new population of solutions is generated. As many new generations are produced the individuals in later generations will differ considerably from those of the initial generation but as a result the average fitness should have increased. This is because only the best individuals from the first generation would have propagated or have been selected for breeding. A "child" solution can be any of the three following types: an *elite child* which is the individual (or indfividuals) of the current generation with the best fitness value and is automatically propagated to the next generation; a *crossover-child* which is created by a combination of a pair of "parent" solutions; and a *mutation-child* which is created by randomly changing (mutating) a current generation individual. This is illustrated in Figure 2.6.

4. Termination: The steps of selection and reproduction are repeated until a termination condition has been satisfied. Usually, such conditions occur when an optimum solution has been found, when the number of maximum generations has been exceeded, when the allocated time or computation budget has been reached, if there is no significant improvement in the fitness of a number of subsequent populations (stall), or because of manual intervention, or any combination of the above.

The pseudocode of a typical GA is given in Algorithm 3 in the appendices. GAs have been applied to the solution of a variety of problems in computer vision such as feature selection [Kim et al. (2006)], face detection [Bebis et al. (1999); Xu et al. (2004)] and object recognition [Hill et al. (1992); Bebis et al. (2002)]. GAs have been shown [Goldberg (1989); Holland (1992)] to perform well in problems involving large search spaces. This is because a GA can locate good-enough solutions very early in the optimisation process while spending the remainder of its allocated time/computation budget trying to improve on those solutions. Quite often the improvements are very small in comparison to the time spent optimising. This is not unusual for other evolutionary methods, some of which we will examine later. That is why we believe that evolutionary optimisation in general may benefit from the inclusion of a local search function after the most productive part of the global search has been carried out.

Differential evolution

Differential evolution (DE) [Storn and Price (1997)] is an evolutionary population-based optimisation algorithm that works on real-valued coded individuals. DE is capable of handling non-differentiable, non-linear and multi-modal objective functions. As with all evolutionary methods, DE maintains a population of candidate solution called the *individuals*. In DE the individuals are represented simply as vector-valued entities. This allows for easier representation of the system variables and handling of objective functions that contain a mixture of discrete, integer and continuous parameters.

The basic way that DE works is that it adds the weighted difference between two randomly chosen population vectors to a third vector and the fitness result is compared with an individual from the current population. In this way, no separate probability distribution is required for the perturbation step and DE is completely self-organising. For example, DE can deduce the perturbation information from the distances between the vectors that comprise the population. At the beginning (exploratory stage) we get a large vector perturbation in order to explore as broad an area as possible. Later on when we are approaching the optimum the distance between the vectors automatically gets smaller and so the perturbations become smaller. This way, DE can carry out a fine grained search for the optimum.

The algorithm behind DE is very simple and works as follows. First we generate an initial population of N individual candidate solution vectors. If there is no prior knowledge about the location of the global optimum we initialise the first population with random values from the known or expected limits of the system variables (boundary constraints). Then for each individual $\overrightarrow{x}_{i,G}$ in the current generation G DE generates a new vector $\overrightarrow{x}'_{i,G}$ by adding the weighted difference between two randomly selected individuals $\overrightarrow{x}_{r1,G}$ and $\overrightarrow{x}_{r2,G}$ to a third randomly selected vector $\overrightarrow{x}_{r3,G}$. The new vector $\overrightarrow{x}'_{i,G}$ is then crossed-over with the original individual $\overrightarrow{x}_{i,G}$ to produce a *trial vector* $\overrightarrow{u}_{i,G+1}$. The fitness of $\overrightarrow{u}_{i,G+1}$ is then compared with that of the original individual $\overrightarrow{x}_{i,G}$. If the fitness of $\overrightarrow{u}_{i,G+1}$is greater than the fitness of $\overrightarrow{x}_{i,G}$ then $\overrightarrow{x}_{i,G}$ is replaced by $\overrightarrow{u}_{i,G+1}$, otherwise $\overrightarrow{x}_{i,G}$ survives in the new population as $\overrightarrow{x}_{i,G+1}$. A more concise pseudocode for a single generation loop can be seen in Algorithm 4 in the appendices, where F is the weighting factor.

In differential evolution, just as in every other evolutionary strategy there are two separate mechanisms that play a central role in the way that the overall population evolves and determine the characteristic behaviour of the optimisation algorithm. The first mechanism is the population's tendency to expand and explore the optimisation landscape. In DE, because of the way new trial vectors are generated, there is a high probability that perturbations yielding acceptable new points will enlarge the search region that is covered by the population and thus prevent premature convergence. The second mechanism is the selection process and is important because of the way it removes vectors in unproductive regions thereby counteracting the continuous expansion of the first mechanism. If left unchecked, the expansion mechanism would cause the population to continue to expand and therefore increase the diversity of the population and diverge to regions which are not of interest. By including the selection process we avoid this problem while ensuring there is enough diversity to explore new territory and make sure that the population is still evolving, thus avoiding a population stagnation.

SOMA

Finally, we examine the Self-Organizing Migrating Algorithm (SOMA). SOMA is a stochastic optimisation algorithm that is modelled on the social behaviour of co-operating intelligent individuals and was chosen because it has been proven that the algorithm has the ability to converge towards the global optimum [Zelinka (2004)]. SOMA was successfully tested on various examples like real-time plasma reactor control [Nolle et al. (2005); Zelinka and Nolle (2004)], deterministic chaos control [Zelinka (2006)] and genetic programming on artificial ant trajectory synthesis [Oplatkova and Zelinka (2006)].

SOMA maintains a population of candidate solutions in every iteration, the latter called a *migration loop*. The initial population is generated randomly inside predetermined boundaries of the solution space at the beginning of the search. In every subsequent migration loop the whole population is evaluated and the individual with the highest fitness (or lower error value) is designated as the leader L (Fig. 2.7(a)). The remaining individuals will "migrate" towards the leader, that is, travel in the solution space at the direction of the fittest individual (Fig. 2.7(b)). The normalised distance travelled by each individual is called the *path length* which is of defined size and is randomly perturbed.

Mutation, as he have seen already in the GA, is the random perturbation of individuals in a population and plays the important role of maintaining the diversity amongst the individuals. Mutation is somewhat different in SOMA than in other evolutionary strategies. SOMA uses a parameter called *PRT* to perturb the individuals and is defined in the range $[0, 1]$. This parameter is then used to construct a perturbation vector (*PRTVector*) as follows:

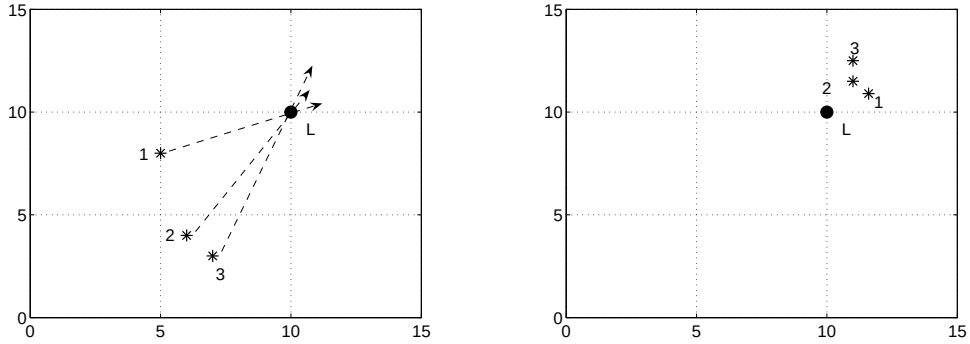
$$\textbf{if } \text{rand}_j < \text{PRT } \textbf{then } \text{PRTVector}_j = 1 \textbf{ else } 0, \;\; j = 1, ..., N,$$

where $rand$ is a random value from $U(0, 1)$ and N is the number of dimensions. The PRTVector determines the final position of a non-leading individual and essentially controls the dimensionality of each individual's movement in the search space. For example, if an element of the perturbation vector is set to 0, the individual is not allowed to change its position in the corresponding dimension.

In most evolutionary methods the *crossover* operation usually creates new individuals based on the information from the existing and previous generations. In SOMA a series of new individuals are obtained with a special crossover operator which in turn determines the movement of an individual in the solution space and thus the overall behaviour of SOMA. The crossover operator is defined as:

$$\overrightarrow{x} = \overrightarrow{x_0} + \overrightarrow{m}\, t\, \text{PRTVector}, \tag{2.5}$$

where $\overrightarrow{x}$ is a new candidate solution, $\overrightarrow{x_0}$ is the original individual, m is the difference between the leader and the starting position of the individual, $t \in [0, \text{PathLength}]$ and PRTVector is the perturbation control vector. We can observe from (2.5) that the PRTVector causes an individual to move toward the leader in $N-k$ dimensions. This is because the N elements of the PRTVector are randomly set to either 0 or 1 and therefore the parameters of an individual will not change in the dimension where $\text{PRTVector}_j = 0$. If we denote by k the number of unchanging parameters, that is the number of dimensions that are not taking part in the actual search process, we can see that the optimisation takes place in $N-k$ dimensional space,



(a) Individuals before migration (b) After migration

Figure 2.7: 2-D examples of the SOMA algorithm.

with at most N dimensions. Such a property can reduce significantly the time SOMA spends searching for a solution. The SOMA algorithm is shown in pseudocode in Algorithm 5 in the appendices.

2.7 Active Appearance Models

Active Appearance Models (AAM) originally proposed by [Cootes et al. (2001)], belong to the general class of linear shape and appearance models and are aimed at solving, among other things, the pose-invariant object recognition problem. AAMs are a very well known and established method that has been used extensively in the past [Edwards et al. (1998); Mitchell et al. (2001); Beichel et al. (2005); Cho and Kim (2007)].

An AAM is a matching technique that combines a parametrised statistical model of the shape and grey-levels[4] of the object and an estimate of the statistical relationship between model parameter errors and resulting image residuals. The AAM is defined by a set of landmarked images that compose the off-line training set. Landmarks are chosen on each training image at key points, such as discontinuity boundaries and feature points, in a similar manner to that we used to landmark the basis views in the LCV training step. In fact, for the AAM tests we have used precisely the same landmark positions to build the appearance models, as we did for the LCV approach. This further facilitates the direct comparison between the two methods, since we are dealing with models of the same shape. Where AAMs and LCV differ, is how the grey-level information is modelled and the combined appearance variation is expressed.

Given thus a set of such landmark points, an AAM is able to generate a statistical model of the shape variation. This is achieved by alignment of all the shape sets from all the training images, into a common coordinate frame (e.g. by using Procrustes alignment [Goodall (1991); Gower (1975)] and carrying out principal component analysis (PCA) [Joliffe (1986)] on the data. Any example of a trained object may therefore be approximated by:

$$x = \overline{x} + P_s b_s, \tag{2.6}$$

[4]We would like to make the distinction between shape appearance, gray-level appearance and combined shape + gray-level object appearance. In this thesis we shall be using the terms shape, gray-levels and appearance to refer each of the object's visual properties respectively.

where, $\overline{x}$ is the mean aligned shape, P_s a set of orthogonal modes of variation and b_s is a set of shape parameters.

For the grey-levels g, in a manner similar to that used in the LCV, a triangulation defined on the landmark points is used. In this case however, each training example is warped to the mean shape $\overline{x}$, and the pixel information g_{im} is sampled over the region covered by the mean shape. In this manner, the object is segmented from the background and only the foreground pixels are used for modelling. The effects of global lighting variation may subsequently be minimised by normalising the resulting samples using a simple affine transformation, and attempting to match each sample to the mean of the normalised data $\overline{g}$, which in itself is an iterative process. It is now possible to apply PCA to the normalised appearance data and obtain:

$$g = \overline{g} + P_g b_g, \tag{2.7}$$

where P_g is a set of orthogonal modes of intensity variation and b_g a set of grey-level parameters.

The combined shape and grey-level appearance of any modelled object may be reached using the vectors b_s and b_g. Since there may be some correlations between the shape and grey-level variations, an additional PCA is carried out on the appearance data:

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix}, \tag{2.8}$$

where W_s is a matrix of weights[5] used to cater for the differences in units between the shape and intensity models. The end result is the combined model, which is given by:

$$b = Qc, \tag{2.9}$$

where Q are the appearance eigenvectors (or orthogonal modes of appearance variation) and c the eigenvalues (or appearance parameters), that control both the shape and grey-levels of the object. As such given a set of parameters c, an example image of a modelled object may be generated by first creating the *shape-free* grey-level image g using:

$$g = \overline{g} + P_g Q_g c, \tag{2.10}$$

and warping it by means of the landmark points defined by:

$$x = \overline{x} + P_s W_s Q_s c, \tag{2.11}$$

where $Q = \begin{pmatrix} Q_s \\ Q_g \end{pmatrix}$.

The final component of the AAM is the active search, where given an appearance model, a novel

[5]The choice of weights is determined by using a displacement-and-error-test methodology [Cootes et al. (2001)], similar to our approach in section 6.3.2.

image and good initialisation, the parameters c are iteratively adjusted in such a way, that in the end, the model matches the novel image as closely as possible. The matching is achieved by minimising the difference $\Delta = |I_i - I_m|^2$ of the grey-scale values in the image I_i and those in the model I_m. This matching step is decoupled from the AAM and indeed any kind of search method may be employed here. [Cootes et al. (2001)] initially proposed a local search method, which is in fact the one we used to evaluate the AAMs on the three datasets. This local optimisation approach makes the search fast and accurate, *provided* a very good initialisation is available close to the global minimum. [Cootes et al. (2001)] do not attempt to solve the general optimisation search over a high dimensional space every time the model is required to fit to the image. Instead they exploit the fact that the optimisation problem is similar each time and that the similarities can be learned off-line, as long as the required object and scene properties have been sufficiently sampled by the training set.

[Cootes et al. (2001)] assume a simplistic linear relationship as an approximation[6] between the change of the model parameters and the error Δ, in order to aid optimisation efficiency. The learning process, during offline training, involves randomly perturbing the model and calculating the error Δ from ground truth images. Once enough such perturbations have been performed, multi-variate regression is used to obtain the parameters of the linear model A.

The pseudocode behind the active search method for a single model search-update iteration, and assuming that the current estimate of the model parameters is c_0, is given in the appendix section in algorithm 6.

The above steps are repeated until the error minimisation is stalled or after some predetermined number of iterations where convergence is assumed. [Cootes et al. (2001)] also use a multi-resolution pyramid search method to achieve convergence at each level before moving on to higher, finer resolutions. This is more efficient than single resolution search when local optimisation methods are used. Tests we carried out using the pyramid approach on global or hybrid optimisation methods for the LCV, did not indicate any better accuracy performance than single resolution alternatives. The only potential advantage of using the pyramid search with a global method on coarser levels is that the latter can be much faster since the image is smaller.

[6] It is only linear over a limited range of values and thus perturbations must be kept low (e.g. $\pm$ 2 pixels translation and 10% scale variation). [Matthews and Baker (2004)] show that this assumption is not correct.

Chapter 3

Background theory

In this chapter we will look into the background theory of multi-view geometry, leading to the formulation of the linear combination of views approach which is an essential part of this thesis. This theory will explain why it is possible to synthesise novel views using 2-D information alone and without the need to recover the 3-D structure of the object.

3.1 Single view geometry

We begin with the simple case of a general projective, pinhole camera model with focal length f and the projection centre placed at the origin of the world frame (Fig. 3.1).

A 3-D world point $P = (X, Y, Z)$ is projected onto the image plane Π through a line that passes from the optical centre C, and is mapped to the 2-D image point p with coordinates given by:

$$x = \frac{fX}{Z}\ ,\ y = \frac{fY}{Z}. \tag{3.1}$$

Equation (3.1) is non-linear. However, if the world and image points are represented using homogeneous

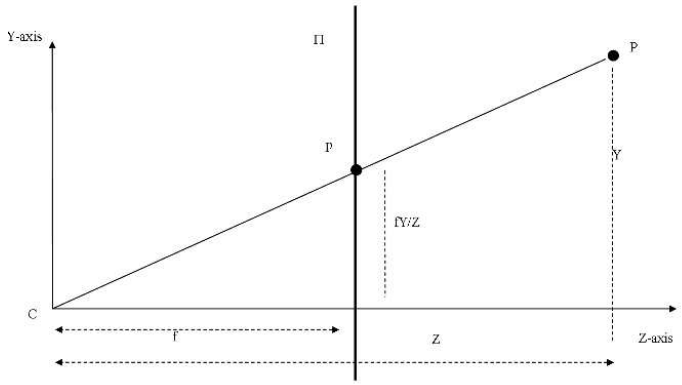


Figure 3.1: Pinhole camera geometry showing the projection of a point P to the image plane Π.

coordinates, then the projection can be expressed linearly in matrix form as:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \approx \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{3.2}$$

The $\approx$ sign implies that the left and right hand sides are equal up to a non-zero scale multiplication. (3.2) can be simply rewritten as:

$$Zp = KP, \tag{3.3}$$

which is known as the *general projective equation*, with $P = [X, Y, Z, 1]^T$ and $p = [fX/Z, fY/Z, 1]^T$. The matrix K in (3.2) represents a very simplistic case since it contains only information about the focal length f. More generally, K is a 3×4 matrix with 11 d.o.f. defined up to a scale factor $\lambda \neq 0$, since K and λK describe the same camera that may be decomposed as follows:

$$K = C\Gamma G. \tag{3.4}$$

Taken in turn, the 4×4 matrix $G = \begin{bmatrix} R_{[3\times3]} & t_{[3\times1]} \\ 0_{[1\times3]} & 1 \end{bmatrix}$ represents the position t and orientation R of the camera with respect to the world coordinate system. These 6 parameters are called the *external* camera parameters. Matrix $\Gamma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ performs the projection from homogeneous world space to homogeneous image space. Finally, matrix $C = \begin{bmatrix} f/s_x & f/s_y \cot\theta & o_x \\ 0 & f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$ is the *camera calibration matrix* which performs a 2-D affine transformation of the image plane and depends on the *intrinsic* camera parameters: focal length f, principal point (or image centre) coordinates (o_x, o_y), pixel width s_x and height s_y and angle θ between the axes (usually $\pi/2$). The ratio $s_x/s_y \approx 1$ is the aspect ratio. If these are parameters are known, the camera is said to be calibrated.

The projective camera equation (3.3) is a non-linear transformation from world to image coordinates which complicates further analysis. To avoid this, we can use one of the available approximations to the projective/perspective camera (Fig. 3.2). The most basic case is the orthographic camera:

$$K_{\text{ortho}} = C \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} G, \tag{3.5}$$

which reduces to a mere parallel projection onto the image plane. However, the orthographic camera is overly simplistic since it does not model the effects of *distance* (i.e. the image of an object will change

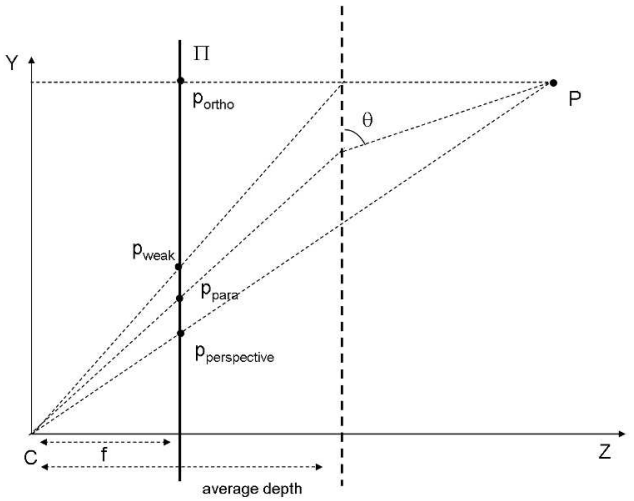


Figure 3.2: Common approximations to the perspective camera.

size as the object's distance from the camera is varied) and *position* (i.e. the image of an object will change as its position in relation to the optical axis is varied). We can approximate the former effect using the average depth $\bar{Z}$ of the scene points in equation (3.1), yielding:

$$x = \frac{fX}{\bar{Z}} \ , \ y = \frac{fY}{\bar{Z}}. \tag{3.6}$$

In matrix form we have:

$$K_{\text{weak}} = C \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{[3\times3]} & t_{[3\times1]} \\ 0_{[1\times3]} & \bar{Z} \end{bmatrix}. \tag{3.7}$$

This is called the weak perspective camera and it is simply the perspective camera with individual point depths Z replaced by an average constant depth $\bar{Z}$ (see Fig. 3.2). The matrix K_{weak} includes two stages: parallel projection onto the average depth plane and uniform scaling of the resulting projection. The weak perspective model is valid when the average variation of the depth of the object $\Delta\bar{Z}$ along the line of sight is small compared to the $\bar{Z}$ and the field of view. As such, K_{weak} does not model position effects leading to a poor approximation when the object is far from the optical axis.

We thus consider an alternative approximation, the para-perspective camera K_{para} where the projection is performed on an arbitrary direction, usually the ray linking the optic centre to the 3-D centroid of the object, which is consistent for all the points. K_{para} can be written as:

$$K_{\text{para}} = C \begin{bmatrix} 1 & 0 & -\cot\phi & \cot\phi \\ 0 & 1 & -\cot\theta & \cot\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{[3\times3]} & t_{[3\times1]} \\ 0_{[1\times3]} & \bar{Z} \end{bmatrix}. \tag{3.8}$$

All three approximations: (K_{ortho}, K_{weak}, K_{para}) can be considered as special cases of the *affine*

camera model which is obtained by constraining K such that:

$$K_{\mathrm{A}} = \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \\ 0 & 0 & 0 & K_{34} \end{bmatrix}, \tag{3.9}$$

thereby reducing the degrees of freedom from 11 to 8. In terms of image and world coordinates, the mapping takes the form:

$$x = AX + t, \tag{3.10}$$

where A is a general 2×3 matrix with elements $A_{ij} = K_{ij}/K_{34}$ and t is a general 2-vector representing the image centre. Although K_{A} is not specified in terms of a decomposition like that given in equation (3.4) it can account for the following: a 3-D affine transformation between world and camera coordinate frames, a parallel projection onto the image plane and a 2-D affine transformation of the world coordinates. We should note here that a collection of homogeneous image points obtained by K_A will have the same projective depths (which by extension also applies to K_{ortho}, K_{weak} and K_{para}) which are independent of the scene structure [Zisserman (1992)].

3.2 Multi-view geometry

We can now move to the geometry of multiple points in multiple views. For this we assume a 3-D scene comprised of a multiple-point vector $[P_1, P_2, \ldots, P_n]^T$. A particular 2-D view of the scene, associated with one camera matrix (e.g. K) may be defined as:

$$S = [KP_1, KP_2, \ldots, KP_n]_{[3\times n]} = K_{[3\times 4]}\,[P1, P2, \ldots, PN]_{[4\times n]}\,. \tag{3.11}$$

According to [Tomasi and Kanade (1992)], S can in principle be factored into two components representing 'joint projection' and 'shape', and its rank is at most 4, which happens to be the least dimension of the factors.

On the assumption that we take a series of images V of the scene each associated with a camera matrix $[K_1, K_2, \ldots, K_V]$ we get:

$$\begin{aligned} S_V &= \begin{bmatrix} K_1P_1 & K_1P_2 & \ldots & K_1P_n \\ K_2P_1 & K_2P_2 & \ldots & K_2P_n \\ \vdots & \vdots & \vdots & \vdots \\ K_VP_1 & K_VP_2 & \ldots & K_VP_n \end{bmatrix}_{3V\times n} \\ &= \begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ K_V \end{bmatrix}_{3V\times 4} \begin{bmatrix} P_1 & P_2 & \ldots & P_n \end{bmatrix}_{4\times n} = \text{Joint projection} \times \text{shape}. \end{aligned} \tag{3.12}$$

Each $3 \times n$ row vector in the S_V matrix contains all the x, y coordinates and the projective depths Z.

We may now consider the transpose of S_V written using the inhomogeneous image coordinates $p = (\frac{x}{Z}, \frac{y}{Z})$ as:

$$S_V^T = \begin{bmatrix} p_1^{'T} & p_1^{''T} & p_1^{'''T} & \dots \\ p_2^{'T} & p_2^{''T} & p_2^{'''T} & \dots \\ \vdots & & & \\ p_n^{'T} & p_n^{''T} & p_n^{'''T} & \dots \end{bmatrix}_{n \times 2V} , \tag{3.13}$$

where $p^{'}, p^{''}, p^{'''}, \dots$ represent the first, second and third views of a point p. If we use the terminology from [Shashua (1997)] each column of S_V^T is part of the Joint Point Space (JPS). Furthermore, each "semi-view" (collection of all x and y coordinates from all points) is inside the column space of S_V^T, the latter being a subspace of the JPS. As a result it should be possible to represent all the views inside the column space provided we construct the appropriate linearly independent basis for it. By definition, the dimension of the column space of S_V^T is equal to the rank of S_V^T, which as we have mentioned previously under affine imaging is at most 4.

3.3 Linear combination of views

[Ullman and Basri (1991)] were the first to point out that we only require three semi-views to span the column space of S_V^T, although as shown in [Buxton et al. (1998)] four semi-views (i.e. 2 views) might be preferable from a practical viewpoint as it results in a symmetric manipulation of the subspace and improved numerical properties in the basis views (e.g. for the solution of the linear system and recovery of the coefficients). More specifically, [Ullman and Basri (1991)] showed that under the assumption of orthographic projection and 3-D rigid transformations, two views are sufficient to represent any novel view of a polygonal object from the same aspect. The proof may easily be extended to any affine imaging condition. Thus, to a good approximation, given two images of an object from different (basis) views I' and I'' with corresponding image coordinates (x', y') and (x'', y''), we can represent any point (x, y) in a novel, target view I_T according to, for example:

$$\begin{aligned} x &= a_0 + a_1 x' + a_2 y' + a_3 x'' + a_4 y'' \\ y &= b_0 + b_1 x' + b_2 y' + b_3 x'' + b_4 y'' \end{aligned} . \tag{3.14}$$

The target view is reconstructed from the above two equations given a set of valid coefficients (a_i, b_j). Provided we have at least 5 corresponding landmark points in all three images (I_T, I', I'') we can estimate the coefficients (a_i, b_j) by using a standard least squares approach. (a_i, b_j) are functions of the camera parameters but without any dependence on 3-D world coordinates. Based on a method for weighting the combination of the intensities (or colours) of corresponding points in the basis views I' and I'' several others have taken this concept further from its initial application to line images and edge maps, to the representation of real images I_T [Bebis et al. (2002); Koufakis and Buxton (1998b); Hansard and Buxton (2000b); Peters and von der Malsburg (2001); Revaud et al. (2007)].

Such results suggest a straightforward yet powerful framework for object recognition: novel views

of an object can be recognised by simply matching them to combination of a small number of stored views (basis views) of the object. The main problem with this idea is the choice of parameters for the combination scheme. As suggested by Ullman and Basri the parameters can be recovered either by: i) identifying a set of features from the novel view that approximately match a set of features from the known views or ii) searching the space of parameters explicitly. In i) one has to compute the transformation that aligns the model with the scene by solving a system of linear equations similar to (3.14). The problem here is the correspondence problem because even under the unrealistic assumption that the correct features have been detected, the number of model-scene feature matches grows exponentially as the number of scene features increases. Techniques that aim to solve this problem, such as the interpretation tree [Grimson and Lozano-Perez (1986)], will be overwhelmed by the sheer number of possible correspondence matches. Strategy ii) avoids this feature-matching step and the correspondence problem but may be very time consuming owing to the high dimensional space that needs to be explored.

In this thesis, we will attempt to use the LCV method directly on intensity images, without extracting any features, establishing correspondences or solving for the LCV coefficients. Instead, we will have to search the high-dimensional parameter space to recover the coefficients with the help of a good and efficient optimisation algorithm. In this context, employing LCV for object recognition has several advantages over existing methods. First, it is more practical than methods which require explicit 3-D models. In fact, a sparse set of 2-D views may be all that is required to represent a 3-D object, and the scheme is as powerful as using 3-D models. Second, it is more efficient since it stores and manipulates 2-D views only. In contrast to multi-view approaches, novel views in LCV are compared to *predicted* views (i.e. combination of reference views) rather than the comparison being the reference views themselves. Since the predicted views can be different from the reference views, recognition does not depend on close similarity between novel and reference views as in the case with multi-view approaches.

Chapter 4

2-D object recognition

In this chapter, we introduce the work carried out during our initial research on the object recognition problem for 2-dimensional objects. The solution we propose consists of a *prototype model template* which describes the representative appearance of a class, a set of *parametric transformations* that deform the template and a set of *constraints* that bias the choices of possible deformations. We begin with basic deformations and continue with their extension and the introduction of probabilistic constraints to build a Bayesian framework. This led us, in addition, to explore the basics of foreground/background modelling and its effect on the template matching process.

4.1 Model representation

The starting point of our investigation into 2-D object recognition is a simple representation for a flat, planar object. We will introduce parametric transformations of such an object later on, but as we wish to avoid additional parameterisation in the model [Cootes and Taylor (2004); Cootes et al. (2001, 1995)] these will only represent global information on the object without explicitly defining a parametric form for each class of objects.

Instead, to aid simplicity, we are going to use a "prototype template", I_m, which is essentially the exemplary appearance or 'model' of an object (or class of objects) and is based on our prior knowledge about the characteristics of the object of interest. Our template thus contains only grey-level and boundary information in the form of a bitmap and is therefore appropriate for general object recognition tasks since, in order to apply the same approach to a different class of objects, we only need to generate a new prototype image of this class. The prototype is usually obtained from training samples, using a training procedure that could be based on Principal Components Analysis (PCA) [Cootes and Taylor (2004)], shape alignment [Viola and Wells (1995); Larsen and Eiriksson (2002); Liang et al. (2006)], or the prototype template could simply be the mean appearance of the class.

If we now revisit the problem statement in section 1.1, we may reformulate (1.1) using the prototype intensity template. We assume a scene or 'target' image $I_T(x, y)$ where $(x, y)^T$ are pixel coordinates. If we allow for a transformation T of the template, our aim is to minimise the difference between the pixel values in the template $I_m(x, y)$ and those in the image $I_T(x, y)$ using, for example, a sum of squares error criterion $\sum_{x,y} [I_T(x, y) - TI_m(x, y)]^2$. The most simple choice for the transformation T is the 2-

dimensional translation of the co-ordinates (x, y) that positions the centre of the template, say, at (u, v). If we also restrict the comparison of target image and template to the area covered by the template, we may thus reformulate (1.1) as:

$$\sum_{x,y} h(x-u, y-v)[I_T(x,y) - I_m(x-u, y-v)]^2, \tag{4.1}$$

where the window function $h(x-u, y-v)$ restricts the sum to be over all the pixels (x, y) under the template located at (u, v). We can now expand (4.1) and obtain:

$$\sum_{x,y} h(x-u, y-v)[{I_T}^2(x,y) - 2I_T(x,y)I_m(x-u, y-v) + {I_m}^2(x-u, y-v)]. \tag{4.2}$$

In (4.2) the term $\sum {I_m}^2(x-u, y-v)$ is constant and if we assume that $\sum {I_T}^2(x, y)$ does not fluctuate very much over different regions of interest $h(x-u, y-v)$ of the target image, we may replace minimisation of the sum of squared differences in (4.2) by maximisation of the *cross-correlation* or *overlap* term:

$$O(u, v) = \sum_{x,y} I_T(x, y)I_m(x-u, y-v), \tag{4.3}$$

which is a similarity measure between the target image and the template. However, strictly speaking, $\sum h(x-u, y-v){I_T}^2(x, y)$ is not approximately constant across the image, especially when there is clutter in the background, but varies with the position of the window $h(x-u, y-v)$. It is thus possible for matching using (4.3) to fail to give consistent results. In particular, this can happen when the correct position where the object is located returns a lower correlation value than, say a bright region in the image where there is a high intensity in $I_T(x, y)$.

We can avoid this particular problem by normalising the intensities of both the target image and the template to unit energy or 'length' by replacing (4.3) with:

$$c(u, v) = \frac{\sum_{x,y} h(x-u, y-v)[I_T(x,y) - \overline{I}_T][I_m(x-u, y-v) - \overline{I}_m]}{\sqrt{\sum h(x-u, y-v)[I_T(x,y) - \overline{I}_T]^2 \sum h(x-u, y-v)[I_m(x-u, y-v) - \overline{I}_m]^2}}, \tag{4.4}$$

which is called the *normalised cross-correlation* and $\overline{I}_m$ and $\overline{I}_T$ are the means of the template and the portion of the image under the window $h(x-u, y-v)$ respectively.

It is immediately obvious that (4.1) does not have a closed form solution and that it must be minimised numerically, using one of the various numerical optimisation algorithms available. Likewise, since it is a similarity measure (4.4) must be maximised numerically. As we can see from Fig. 4.1(a) however, this is not so straightforward since even an elementary transformation, such as translation of the template, can generate very noisy surfaces replete with local minima and is as a result very difficult to optimise without a complete global search. Such a complete global search is usually carried out in template matching by scanning the template over all possible locations in the target image but this procedure, does not extend well to cases where the transformation T is more complicated and the search is

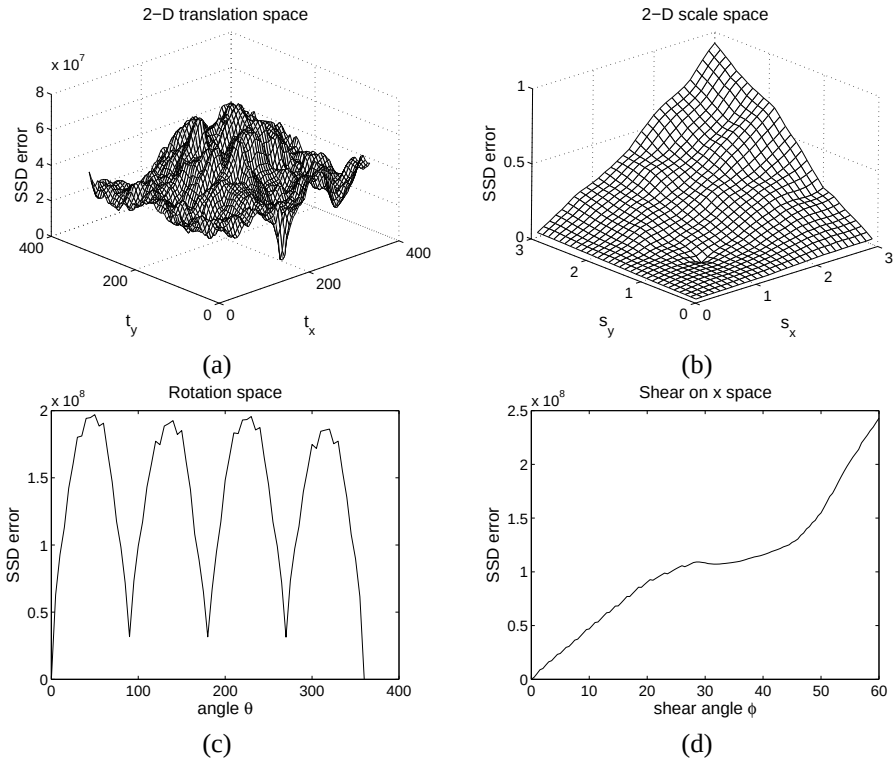


Figure 4.1: surfaces and curves composing the affine transformation.

higher-dimensional. For this reason, we need to simplify the problem by regularising the error surface and/or starting the optimisation process much closer to the basin of attraction[1]. We shall revisit this idea later in this chapter.

4.2 Parametric transformations

Although we assume that the prototype template exhibits the instance of the object that is most likely a-priori, we still need the ability to deform it to match the image. The 2-D translation previously used is very restrictive for most object recognition applications so we would like to extend it to a more powerful transformation, the global affine transformation with 6 degrees of freedom (d.o.f.) which, for example, can be used approximately to account for changes in the apparent shape of a 2-D object with viewpoint. The affine transformation is represented here as: $T = M + d$, where

$$M = \begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix} \tag{4.5}$$

is a 2×2 linear transformation matrix with 4 d.o.f. and $d = (d_x, d_y)$ a translation vector with 2 d.o.f.. These transformations may be the result of variations in the location and shape of the object itself or, as noted above, variations in the camera viewpoint (distance, viewing angle and so on).

If we now try to minimise the dissimilarity between the template and scene image with respect to

[1] In the context of the optimisations we have to carry out, we will loosely define the basin of attraction as the region of space (i.e. set of points) such that initialisation within this region will guarantee convergence of the optimisation algorithm to the global optimum. In this sense, the basin of attraction is algorithm-dependent.

the 6 parameters of the affine transform T, we will soon discover that for the majority of cases (excluding very simplistic objects over constant backgrounds and for trivial differences between scene and template) this is indeed a difficult task that can defeat, for example, pyramid-based matching procedures and even more sophisticated optimisation algorithms. This is because, as noted above, pixel-based template matching usually involves dealing with very complicated error surfaces.

A possible solution to this problem would be somehow to assist the search algorithm. This usually either means initialising the optimisation close to where we believe the solution might be or by incorporating prior information into the optimisation process that will, we hope, constrain the solution towards the desired global optimum. The latter might take the form of a restriction of the search to possible good areas in the parameter space that should be explored or of a regularisation of the error surface by addition of a term or terms which are convex and sufficiently strong to dominate the pixel-based matching term everywhere except near the desired global optimum. Since good initialisation without explicit knowledge of the solution set might not always be possible, we decided to introduce prior information by associating probability distributions with the parameters of the affine transform and building a Bayesian model. To achieve this we need to choose a suitable parameterisation of the affine transformation. Ideally, the chosen representation would isolate the individual degrees of freedom into separate independent transformations and assign a probability distribution to each one. The reason for this is that dealing with statistically independent parameters is both more practical and more intuitive than dealing with multivariate distributions. In particular, we are able to examine independent univariate transformations in isolation and assign to them pdfs chosen specifically for their individual characteristics

It is therefore necessary to *decompose* the linear matrix M as far as possible into individual meaningful transformations (primitive matrices). One way to proceed is via polar decomposition [K.Shoemake and Duff (1992)] and to decompose the (in general) non-singular matrix M as $M = QS$, where Q is an orthogonal matrix with 1 d.o.f. that, depending on the sign of its determinant, may be a pure rotation and S is a symmetric and (in general) positive definite *stretch* matrix (i.e. a non-uniform scale along orthogonal axes that may be turned at an angle to the coordinate axes) with 3 d.o.f.. Polar decomposition will produce unique matrices Q and S. Unfortunately in general this is as far as we can go and we cannot uniquely[2] decompose S any further into scale or shear matrices. In addition the order in which the constituent matrices are multiplied matters, introducing further ambiguity.

Given these difficulties, another way to proceed is to *compose* the linear matrix M as a product of primitive matrices. For example, if we adopt a canonical order for the transformations we can say:

$$M = SRU_x, \tag{4.6}$$

where:

$S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$ is an anisotropic scale matrix with 2 d.o.f. ,

[2]We could try a further polar decomposition on S to obtain a shear matrix but this will not work because of the interaction effects of the shear transformation. See [K.Shoemake and Duff (1992)].

$R = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$ is a rotation matrix with 1 d.o.f. and

$U_x = \begin{bmatrix} 1 & -tan\phi \\ 0 & 1 \end{bmatrix}$ a shear matrix on the x axis with 1 d.o.f. .

As we can see, we have accounted for all the degrees of freedom of the linear matrix M. Of course, this composition is not unique, and indeed any such combination that has 4 d.o.f. will be valid. Since we are only interested in the transformations from an optimisation point of view, the order in which the transformations take place (e.g. shear followed by rotation and then anisotropic scale) should not matter as long as we use the same representation throughout.

Having representing the matrix M in such a way, we may begin by exploring the characteristics of the individual transformations independently from each other, near and inside their respective basins of attraction. In Fig. 4.1 we show the SSD error response for each of these transformations. These were produced by placing a windowed template directly over the imaged object and varying each of the 6 transformation parameters in turn while having conditioned the remaining ones at their optimal values. The results are only 1- and 2-dimensional slices of the overall basin of attraction which owing to the high dimensionality cannot be viewed in its entirety. They are still however very useful in identifying where potentially interesting solutions may exist and helping to choose the appropriate prior distributions.

In addition, though it is not strictly necessary in this affine model, we have chosen to include a local, flexible deformation L which is a continuous mapping $(x, y) \rightarrow (x, y) + [L_x(x, y), L_y(x, y)]$ in 2 dimensions. We define it as a simple sinusoidal wave function:

$$L_\psi(x, y) = [L_x(x, y), L_y(x, y)] = \left[\alpha cos(\frac{2\pi\Delta}{\lambda_x}), \beta cos(\frac{2\pi\Delta}{\lambda_y})\right], \tag{4.7}$$

where $\psi = (\alpha, \beta, \lambda_x, \lambda_y, x_0, y_0)$ are the deformation parameters. α, β are the wave amplitudes, λ_x, λ_y the wavelengths, and $\Delta = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ is the Euclidean distance from the wave centre point (x_0, y_0). Although we assume an affine or weak perspective camera model it is important to consider effects due to image distortion via lens aberration and other non-linear processes during image formation. Such effects may of course be removed by means of a suitable camera calibration [Salvi et al. (2002); Hemayed (2003)], however, un-calibrated cameras are frequently used in practice and this is becoming increasingly the case as cheap digital cameras become widely available. Thus, the wave deformation L is used to introduce any necessary curvature into the mapping process and to take care of fine detailed adjustments that the affine transformation alone cannot explain. Defined in this way the local deformation represents extrinsic variation, but there is no reason why it could not also be used to represent intrinsic shape changes of the model especially if applied before the affine transformation M. The deformation L is similar to the orthogonal base displacement used by Jain et al. [Jain et al. (1996)] but in our case is simpler and easier to optimise.

The function L is continuous and smooth for low values of α, β and λ_x, λ_y approximately the size of the template window and thus maintains the connectivity and smoothness of the template. For higher

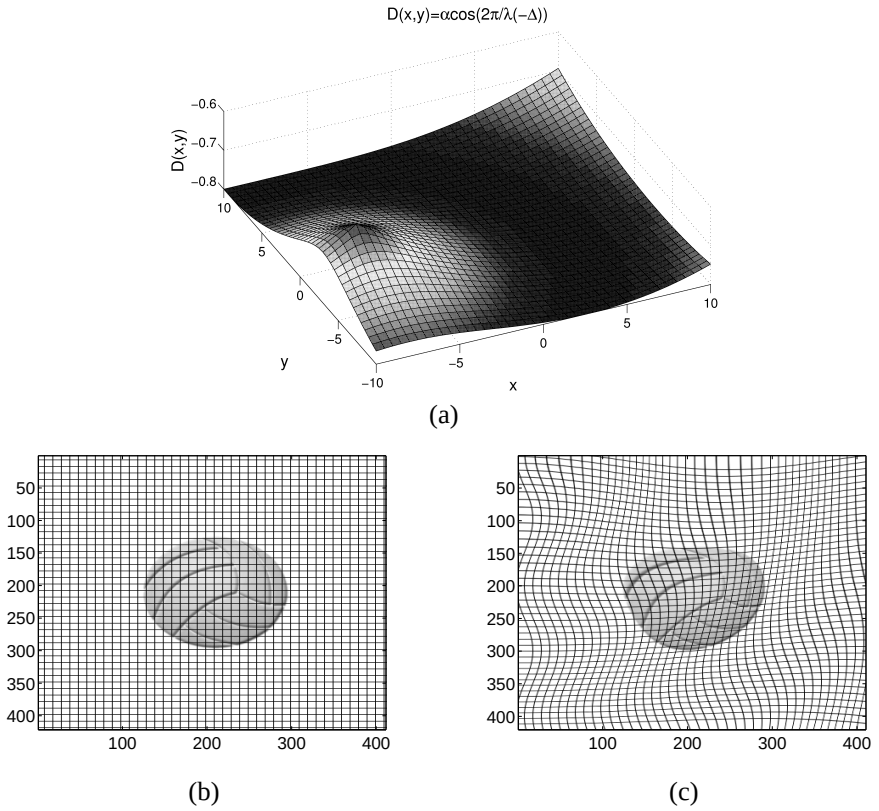


Figure 4.2: The sinusoidal wave (a) and its deformation effects (c) on a 2-D shape (b).

amplitudes and lower wavelength values we can obtain more complex, coarser deformations if required. The wave function has simple parameters that are meaningful and can easily be adjusted to control the wave propagation over the image, and in addition, it is straight-forward to attach probabilistic priors to them. An example of this local deformation function together with its effect on an image can be seen in Fig. 4.2.

Suppose then that we have a prototype template function $I_m(x, y)$ and a transformation T that transforms the template as follows:

$$I_S(x, y) = TI_m(x, y) = I_m([M(x, y)] + L_\psi(x, y) + d). \tag{4.8}$$

If we use (4.6) from above we see that:

$$I_S(x, y) = I_m(SRU_x(x, y) + L_\psi(x, y) + (d_x, d_y)) \tag{4.9}$$

which is the parametric transformation that will deform the template to produce a synthetic image I_S, say, to be matched to the scene or target image I_T. This transformation is realised by shearing the template by an angle ϕ, then rotating by an angle θ, scaling the result by s_x, s_y along directions x and y respectively, locally deforming the resulting template by ψ and finally translating by d.

We can now use equations (1.1) and (4.9) and minimise for the transformation parameters ξ, say, in order to obtain the optimal solution that will match the rectangular template to the image. Since

this is a non-linear objective function we need an iterative method in order to minimise it successfully. Furthermore, because only a limited set of parameters ξ will produce a template that closely resembles the object we may expect a narrow basin of attraction in a high-dimensional space. Unless we initialise the optimisation very close to the solution representing the correct match, minimisation is likely to be difficult. The alternative is to restrict the variability of the transformations known to be likely to represent correctly matching solutions.

4.3 Probabilistic constraints

By choosing appropriate transformation parameters we can represent a large set of possible transformations of the prototype template. However, not all these choices will produce a valid template or even a template that resembles the object(s) in the image. Constraining the choice of possible parameters ξ may thus yield better solutions. We do so by imposing a probability density function (pdf) on the parameters of the transformations T.

Consider the local deformation $L_\psi(x, y)$ first. We have deduced constraints on the range of acceptable parameter values based on experimentation and insight into the transformations with which they are associated. First and foremost, we have chosen a uniform distribution for the wave centre parameters x_0, y_0 since any starting point (within the image range) has an equal probability of producing a valid wave. Under the assumption that the amplitudes α, β of the two waves (one on the x-axis and the other on the y-axis) are zero mean, independent and identically Gaussian distributed, with equal variance $\sigma_\alpha^2 = \sigma_\beta^2 = \sigma_w^2$ then their pdf will be:

$$\begin{aligned} P(\alpha,\beta) = P(\alpha)P(\beta) &= \tfrac{1}{\sigma_\alpha\sqrt{2\pi}} \exp\left\{-\tfrac{\alpha^2}{2\sigma_\alpha^2}\right\} \tfrac{1}{\sigma_\beta\sqrt{2\pi}} \exp\left\{-\tfrac{\beta^2}{2\sigma_\beta^2}\right\}. \\ &= \tfrac{1}{\sigma_w^2 2\pi} \exp\left\{-\tfrac{\alpha^2+\beta^2}{2\sigma_w^2}\right\}. \end{aligned} \tag{4.10}$$

Generally, if we choose large values for α, β we will obtain large deformations of the template and thus large deviations from the original prototype. As we have indicated above we wish to avoid that and we do so by adjusting the variance σ_w^2. Large values of σ_w^2 allow for larger deformations and vice-versa smaller values tend to restrict the parameters to representing smaller deformations. The wavelength parameters λ_x, λ_y require a different pdf with positive or negative non-zero values (multiples of the image width and height respectively) being more probable than wavelengths close to zero. Therefore, it is clear that we need a distribution that is symmetric, with zero probability for when the wavelength $\lambda = 0$, and which increases as we move further away from the origin. Since such a pdf is not easily expressed in a familiar analytic form we have reformulated (4.7) by using wave number parameters $k_x = 1/\lambda_x, k_y = 1/\lambda_y$ each of which is the reciprocal of the wavelength. k_x, k_y have units of inverse length and represent the number of waves (or cycles) per unit distance. The new wave deformation will thus be as follows:

$$L_\psi(x, y) = [\alpha\cos(2\pi k_x\Delta), \beta\cos(2\pi k_y\Delta)]. \tag{4.11}$$

This reformulation allows us to use the parameters k_x, k_y instead of λ_x, λ_y which is preferred because the wave number parameters k_x, k_y have much simpler pdfs. More specifically, the probability $P(k_x, k_y)$ should have a maximum at 0 where each of the deformation wavelengths becomes very small since the template will then undergo only insignificant deformations. The probability may be expected to decrease quickly, for example exponentially, as we move away from 0. Such a pdf has the characteristics of the Laplacian (or double-exponential) probability density function but since its derivative has a discontinuity at zero we decided to approximate the pdf using the much simpler Gaussian distribution. If we again assume that k_x and k_y are independently and identically distributed with means $\bar{k_x}, \bar{k_y} = 0$ and shape parameter σ_k then their pdf is:

$$\begin{aligned} P(k) = P(k_x)P(k_y) &= \tfrac{1}{\sigma_k\sqrt{2\pi}} \exp\left\{-\tfrac{k_x^2}{2\sigma_k^2}\right\} \tfrac{1}{\sigma_k\sqrt{2\pi}} \exp\left\{-\tfrac{k_y^2}{2\sigma_k^2}\right\} \\ &= \tfrac{1}{\sigma_k^2 2\pi} \exp\left\{-\tfrac{k_x^2+k_y^2}{2\sigma_k^2}\right\}. \end{aligned} \tag{4.12}$$

. Note that (4.12) represents our empirical, expectation that waves with smaller wavelengths (thus smoother deformations) are more probable since in generally we do not deal with severe non-linear lens distortions.

For the rotation and translation, we may assume, for example as a default, that all rotations and translations are equally possible and thus we can consider their parameters θ, d as being uniformly distributed. However the scale and shear transformations require a different approach and special care is required in choosing their pdfs. The reason for this comes from the behaviour of the error function (4.9) for certain values or ranges of values for the parameters $s = (s_x, s_y)$ and ϕ. For example, if one or both of the scale parameters are very small, $I_S(x, y)$ will collapse into a single point or to a line respectively. This of course is not going to be a valid representation for the template but the error will undoubtedly have a minimum for these values of the scale parameters. Such trivial solutions should not be allowed. Similar behaviour occurs with the shear angle ϕ.

To illustrate this further we have carried out the following experiment. We took a grey scale template of an object, created directly from an image, and placed that template above the original scene object. Then we sampled the sum of squared differences error function for different values of the scale parameter. We started from $s = 1$ (original template size) and scaled it up until $s = 3$ and we also scaled down the original template until $s = 0$. The resulting error function plot can be seen in Fig. 4.3, upper left image. It is important to note how the error function behaves as we vary the scale parameter. As expected, for a specific value of s (in this case $s = 1$), we have a correct object-template match and the error function is at a minimum. However, we can also see that for values $s < 0.8$ the error function decreases and eventually drops to zero at $s = 0$. In this case, where the template was constructed from part of the image itself, along with the solution at $s = 1$ the solution at $s = 0$ is a global minimum. An optimisation algorithm might correctly identify this as the minimum but such a solution is not useful since the template would have collapsed to a point. For the sear parameter ϕ, a shear defined in terms of the shear angle (as opposed to the shear magnitude) will for $\phi = \pm\pi/2$ collapse the original template

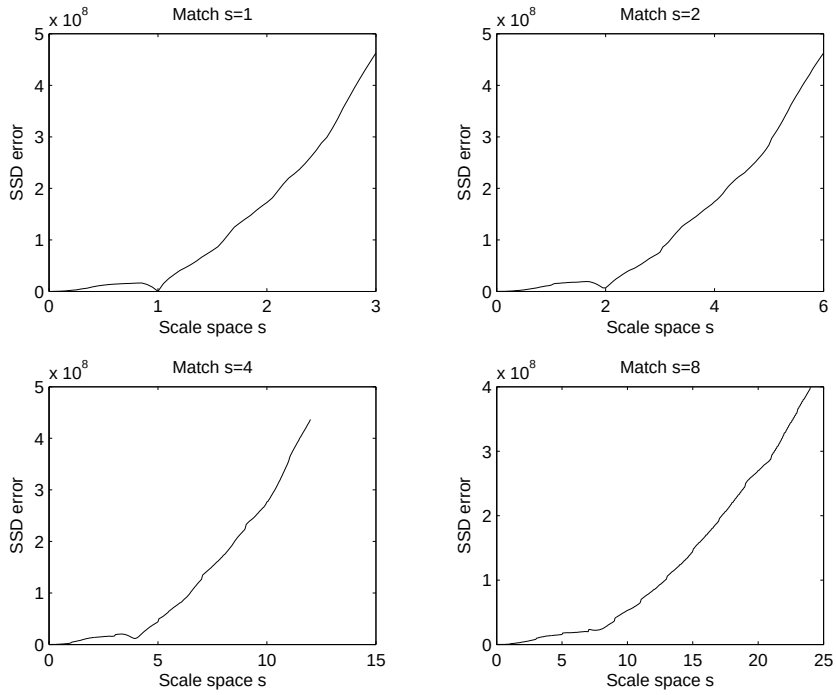


Figure 4.3: Error function appearance for a match located at successively higher scale values.

into a line.

In addition to the behaviour of the error function when the scale parameters are close to zero, another problem arises when the global minimum is located further away on the rightmost part of the scale axis (see Fig. 4.3). It is noticeable that there is some degradation in the quality of the minimum and that as the value of the scale at which it occurs increases the minimum becomes less pronounced. Eventually, for sufficiently large scale values the desired response will completely disappear and thus will be undetectable by any optimisation technique. This behaviour is caused by the fact that when we need to scale-up the template in order to match with the image we have to use an interpolation method. The more we have to interpolate, the more details of the object's appearance may be omitted and the greater the match degradation. This is a problem inherent to the way the prototype is modelled and the type of image to which it is applied. If the prototype were modelled at one scale but the object in the image is at a considerably larger scale then we will have a situation like that described above. From this point of view a solution to this problem is very difficult and care should be taken when building and applying a prototype template so that the match is located within certain limits of the scale of the original template. A hierarchy of templates constructed from training images at different magnifications and/or different viewing distances [3] might thus be used. We note that it is possible to a greater extent to use a prototype template that is at a considerably larger scale than the object we are expecting to find in the image. This will mean that we will have to reduce the scale in order to find a match but, because unlike up-scaling, downscaling does not "invent" new information for the model but instead reduces the

[3]Note that in general changes in magnification and changes in viewing distance are not equivalent because occlusion changes may be associated with the latter but not with the former.

information content to match that of the image, the quality of the match will not be so badly affected (provided of course we carry out the interpolation correctly)[4].

To avoid the problems with the trivial, invalid solution we seek to forbid such problematic values for the scale and shear parameters. For this reason we define a prior for these parameters that will bias them away from such values. From the examples we have seen for the scale parameter it is obvious that we need to impose a distribution that is applicable to random, continuous variables that are constrained not to be zero but may take a few large values. We therefore require a distribution that is asymmetrical and positively skewed, preferably with the possibility of adjusting the tail at large scale. A good choice (as we shall explore later in the next section) for the scale parameters s_x and s_y is the *lognormal distribution* [Evans et al. (2002)]. If we assume that s_x and s_y are independent and their shape and scale parameters are equal $b_x = b_y = b$ and $\sigma_x = \sigma_y = \sigma$ respectively, this choice leads to:

$$P(s_x, s_y) = \frac{1}{s_x s_y b^2 2\pi} \exp\left[-\frac{(\log(s_x) - \sigma)^2 + (\log(s_y) - \sigma)^2}{2b^2}\right]. \tag{4.13}$$

The lognormal distribution assigns very low probability to quantiles close to zero while it allows us to determine the probability of large values of the scale parameters s_x, s_y by adjusting the tail of the pdf.

For the shear angle, we would like to introduce a bias in favour of small deformations and that specific values close to integer and a half multiples of $\phi = \pm\pi/2$ are not admitted. In addition, when the mean shear angle $\bar{\phi}$ is at or near to zero the distribution must be symmetric and have a high probability. If on the other hand, the mean angle is close to $\pm\pi/2$ then the probability must fall sharply. It is obvious then that the shape of the pdf must change from symmetric to positively or negatively skewed as we move $\bar{\phi}$ along the shear angle axis. We have therefore chosen a mixture model of two opposite *Gumbel distributions* (extreme value Type 1) [Evans et al. (2002)] with the mixture weight parameters chosen to ensure the following: First, when the mean shear $\bar{\phi}$ is at either of the two extremes of the shear parameter axis (the range of the shear parameter is $-\pi/2 < \phi \leq \pi/2$) only the one of the two Gumbel distributions with the appropriate skewness will contribute; Second, when $\bar{\phi} = 0$ both Gumbel distributions will contribute equally thus enforcing symmetry in the mixture. The pdf of this mixture may be formulated as:

$$P(\phi) = \frac{(1-A)e^{-\frac{\phi-\bar{\phi}}{b} - e^{-\frac{\phi-\bar{\phi}}{b}}} + Ae^{-\frac{\phi-\bar{\phi}}{b} - e^{-\frac{\phi-\bar{\phi}}{b}}}}{b}, \tag{4.14}$$

where b is the shape parameter and $A = \frac{\bar{\phi}+\pi/2}{\pi}$. An illustration of the mixture model can be seen in Fig. 4.4. Since the individual transformation parameters where assumed independent, the total prior pdf is the product of the individual pdfs (4.12), (4.13) and (4.14), $P(\xi) = P(k)P(s_x, s_y)P(\phi)$.

[4] In this case too, if the template is constructed from too small a viewing distance, interpolation will not, in general, correctly represent non-linear occlusion effects.

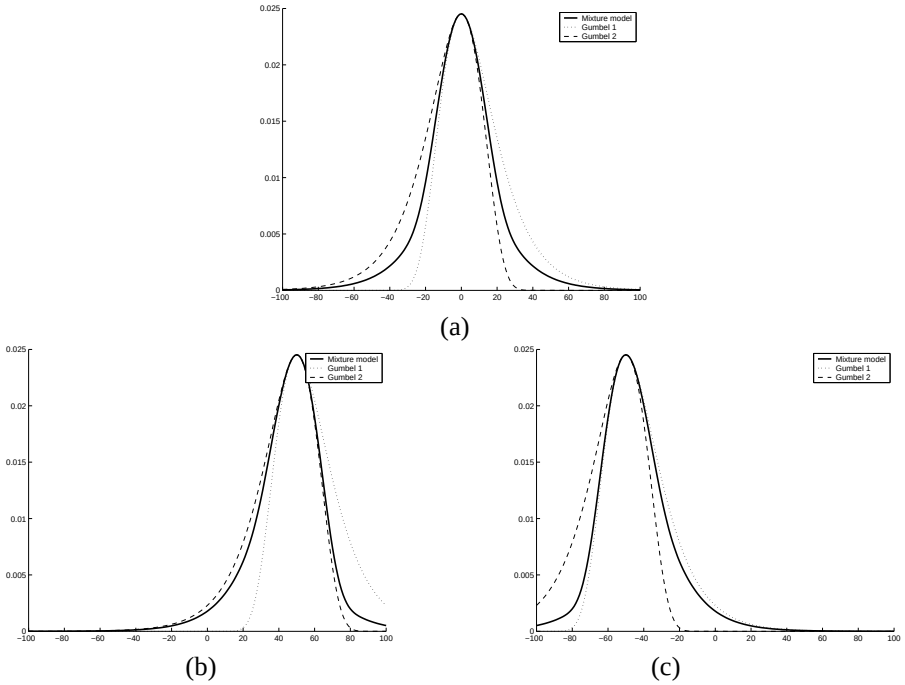


Figure 4.4: The mixture model (bold line) for the distribution of the shear parameter ϕ.

4.4 Objective function

Now that we have established the form of the prior pdf we complete definition of the objective function by means of a Bayesian formulation as described in Chapter 1. If we return to the general object recognition equation (1.1) this means that, having identified the appropriate transformation (4.9) which will deform the prototype template I_m, all we need now is to define a suitable measure g. Two widely used such measures are the sum of squared differences (SSD) or L_2 norm, and the sum of absolute differences (SAD) or L_1 norm, which measure the dissimilarity between the image and the template.

Although the SSD metric has been used in a variety of object recognition problems it is not without serious limitations. First and foremost, SSD is sensitive to outliers and not robust to template variations. Even though it is valid from a maximum likelihood perspective when the template is actually a model of the object of interest in the target image, a SSD metric assumes a normal distribution on the residuals (i.e. the error) and independence on the variables used to derive the likelihoods. However, [Tian et al. (2004)] have shown that additive noise in real images is generally not independently and identically normally distributed. Noise models that are normally distributed usually assume statistical independence of adjacent pixels. Since however in practice the majority of variation in an intensity image is due to illumination changes or to intrinsic variation between similar in-class objects and since such variations are spatially correlated, this assumption is not plausible. Furthermore, the residuals are very different and very differently distributed when the template lies over the object and when it lies over the background. In the former case the residuals may be assumed to be small and due to noise and/or accumulation of modelling inaccuracies. In the later case the residuals will be large and, for an arbitrary template and image background, distributed in the same kind of way that natural imagery is. Since the intensity of an image depends on both the illumination conditions and camera settings and properties, it is difficult

to model its distribution for images of natural or man-made scenes in detail, but the distribution of the outputs of banks of filters applied to such images has been described for example by [Srivastava et al. (2003, 2002); Mumford (1996); Huang and Mumford (1999)]. In addition, as we noted in section 4.1 it has been pointed out by [Sullivan et al. (2001)] that, in a valid Bayesian analysis, our data observations must be regarded as fixed and not as a function of the hypothesis as to what the image template represents, how big and what shape it is, and where it is located in the image. The SSD metric as commonly used violates this principle by considering only the portion of an image directly under the template I_m. Instead, we should incorporate the background information from the image, for example: by sampling the background so that it is known a priori, by choosing it to be very simple, such as a uniform bland image or dark, or by building a probabilistic model of the image background.

Contrary to the SSD metric, the SAD metric is more robust since it does not give such high importance to large residuals. The SAD metric may be justified from a maximum likelihood perspective when the noise distribution is Laplacian. Nevertheless this function is not smooth and is singular when the error residuals approach zero. Such a singularity may cause difficulties in numerical optimisation in particular if gradient-based methods are used. This metric has the advantage that large residuals are given only the same importance or "influence" as smaller residuals.

Thus we require a more robust error measure, one that treats residuals over foreground areas with one metric and residuals over background with a different metric. A first approach is to use one of the $L_1 - L_2$ hybrid norms, such as the one proposed by [Huber (1973)]:

$$g_\tau(I_T, I_m) = \begin{cases} \frac{(I_T - I_m)^2}{2\tau} & , 0 \leq |I_T - I_m| \leq \tau \\ |I_T - I_m| - \frac{\tau}{2} & , \tau \leq |I_T - I_m| \end{cases}, \tag{4.15}$$

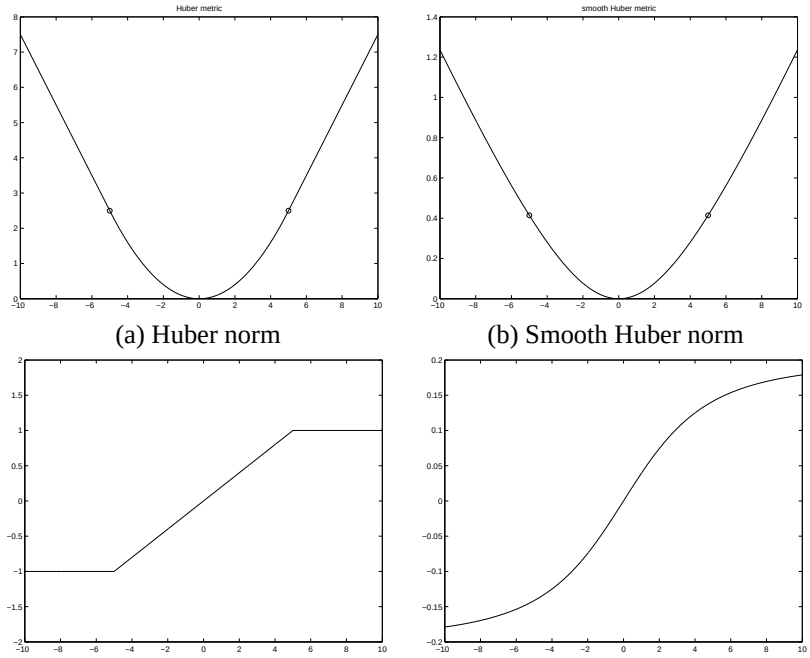
where τ is the threshold at which the function switches between the L_1 and L_2 norms. Fig. 4.5(a) shows the Huber norm as a function of the residuals and how it treats small residuals (between $-\tau$ and τ) with the L_2 norm and large residuals with the L_1 norm. The marks represent the point where we switch from L_1 to L_2 and vice versa. Even though the Huber norm is smooth at $\pm\tau$ where it switches between the two norms it is only C^1 continuous (see Fig. 4.5(c)). One can go further by introducing a metric that smoothly interpolates between the two norms. One such metric is the smooth Huber norm [Buxton (2004)] defined as:

$$g_\tau(x) = \sqrt{1 + \frac{x^2}{\tau^2}} - 1, \tag{4.16}$$

and whose function and first derivative are illustrated on Fig. 4.5(a) and (b) respectively. If we use equations (1.1), (4.9) and (4.16) we obtain the combined objective function which needs to be minimised:

$$\hat{p} = \underset{T}{\operatorname{argmin}} \sum \left(\sqrt{1 + \frac{[I_T(x,y) - TI_m(x,y)]^2}{\tau^2}} - 1 \right), \tag{4.17}$$

where the threshold τ may be chosen at $\tau = \frac{\max|X|}{100}$, or set at the 98^{th} percentile of the observed data X (see [Guitton and Symes (2003); Guitton and Verschuur (2004)]).



(a) Huber norm (b) Smooth Huber norm

(c) Huber 1st derivative (d) Smooth Huber 1st derivative

Figure 4.5: Comparison of the Huber and smooth Huber norms.

Since we are using a Bayesian approach we need to reformulate (4.17) as a pdf. The likelihood of observing the input image given the deformations on the prototype template is therefore:

$$P(I_T|\xi) = C_1 \exp\left\{-\sum\left(\sqrt{1+\frac{[I_T(x,y)-TI_m(x,y)]^2}{\tau^2}}-1\right)\right\}, \tag{4.18}$$

where, as usual ξ stands for the parameters of the transformation T, C_1 is a normalising constant equal to $1/2(eK_1(1)\tau)$, e is the exponential and K_1 a modified Bessel function (using (4) from [Gradshteyn and Ryzhik (1980)], p. 358 and changing variables). C_1 simply ensures that (4.18) integrates to 1.

Finally, we may ignore not only these constant terms (since they do not make much difference from an optimisation point of view) but also use the fact that the probability $P(I_T)$ is constant, $P(\xi|I_T) \propto P(I_T|\xi)P(\xi)$ to combine equations (4.12), (4.13), (4.14) and (4.18) to obtain the posterior pdf of the parameters ξ given an image I_T. The parameters may be recovered by minimising the corresponding negative log-likelihood:

$$\begin{array}{c}\min_\xi\{-\log P(\xi|I_T)\} = \log(\sqrt{s_x^3 s_y^3}) - \log(e^{-\frac{\phi}{b}-e^{-\frac{\phi}{b}}} + e^{\frac{\phi}{b}-e^{\frac{\phi}{b}}}) + \frac{k_x^2+k_y^2}{w^2} \\ +\sigma_s(\frac{1}{s_x}+s_x+\frac{1}{s_y}+s_y-4) + \frac{\alpha^2+\beta^2}{\sigma_{\alpha\beta}^2} + \sum_{x,y}\left(\sqrt{1+\frac{[I_T(x,y)-TI_m(x,y)]^2}{\tau^2}}-1\right)\end{array}. \tag{4.19}$$

Note that the distribution shape parameters $b, w, \sigma_s, \sigma_{\alpha\beta}$ and the threshold τ are treated as fixed.

4.4.1 The scale transformation

We mention the scale transformation here separately because it has some interesting properties that we would like to explore and also because it poses some difficult problems for object recognition systems.

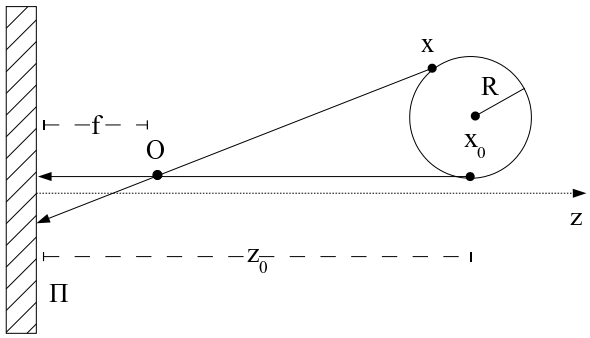


Figure 4.6: A model of the spherical imaged object under a perspective camera model.

In this section we will examine various distribution models to determine which one better describes the process of an imaged object undergoing uniform scaling. This may subsequently be used as a prior in a Bayesian formulation to capture knowledge about the scaling process.

The first step is to define a theoretical model of how the scale of an object changes in relation to the viewing distance. This model is a theoretical analogue of the actual deformation process and can be used to generate prior distributions for the scale parameter, for example, by appropriately sampling the viewing distance parameter. In addition it can help us to understand the distribution of the viewing distance parameter when there is only explicit knowledge of the scaling of object appearance for example after a practical imaging experiment as we will see later on. This in turn can help to verify the correctness of, and any inherent statistical bias in, our experiments. Once the scaling model is defined we are able to fit parametric models of the distribution of scale and determine which one has the best properties to describe our prior knowledge about an object that undergoes a scale transformation. The chosen parametric model may then be used as a prior for the scale parameter in our Bayesian inference paradigm.

Thus, we assume a perspective camera model such as the one illustrated in plan view in Fig. 4.6 and imaging a spherical object defined by the equation $(X - X_0)^2 + (Y - Y_0)^2 + (Z - Z_0)^2 = R^2$. (X, Y, Z) is a point on the boundary of the sphere, (X_0, Y_O, Z_O) is the sphere's centre and R its radius. The camera is defined by the centre of projection O, the imaging plane is Π, the focal length f and the viewing axis z. We denote the distance between the image plane and the object by Z_0. It can be shown that for a perspective camera model the imaged boundary (x, y) of such a spherical object of radius R (the reason for choosing a sphere is for simplicity and will become apparent later on) has the equation of an ellipse:

$$\left(\frac{X_0^2 + Y_0^2 + Z_0^2 - R^2}{X_0^2 + Y_0^2}\right)(xX_0yY_0)^2 + \left(\frac{Z_0^2 - R^2}{X_0^2 + Y_0^2}\right)\left[xX_0 + yY_0 + fZ_0\left(\frac{X_0^2 + Y_0^2}{Z_0^2 - R^2}\right)\right]^2$$

$$= \frac{f^2R^2(X_0^2 + Y_0^2 + Z_0^2 - R^2)}{Z_0^2 - R^2}, \tag{4.20}$$

with centre:

$$x = \frac{X_0fZ_0}{Z_0^2 - R^2}, \quad y = -\frac{Y_0fZ_0}{Z_0^2 - R^2} \tag{4.21}$$

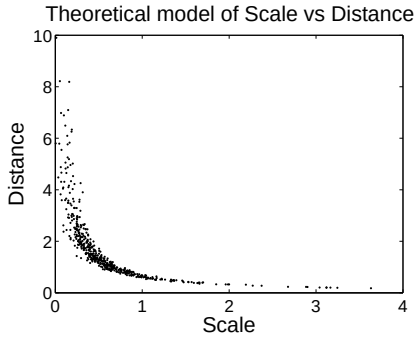


Figure 4.7: Distance vs Scale theoretical process model.

and semi-major and semi-minor axes:

$$a^2 = \frac{f^2R^2(X_0^2 + Y_0^2 + Z_0^2 - R^2}{(Z_0^2 + R^2)^2}, \quad b^2 = \frac{f^2R^2}{Z_0^2 - R^2} \tag{4.22}$$

respectively.

We may now define the scale s of the imaged, elliptical boundary as:

$$s^2 = ab$$

$$\Rightarrow s = \frac{fR}{\sqrt{Z_0^2 - R^2}} \left(\frac{X_0^2 + Y_0^2 + Z_0^2 - R^2}{Z_0^2 - R^2} \right)^{1/4}. \tag{4.23}$$

If we now assume that the object is approximately centred in the image, so $X_0 = Y_0 = 0$ and that its radius is much smaller than the viewing distance $Z_0^2 \gg R^2$ then, without loss of generality, (4.23) simplifies to:

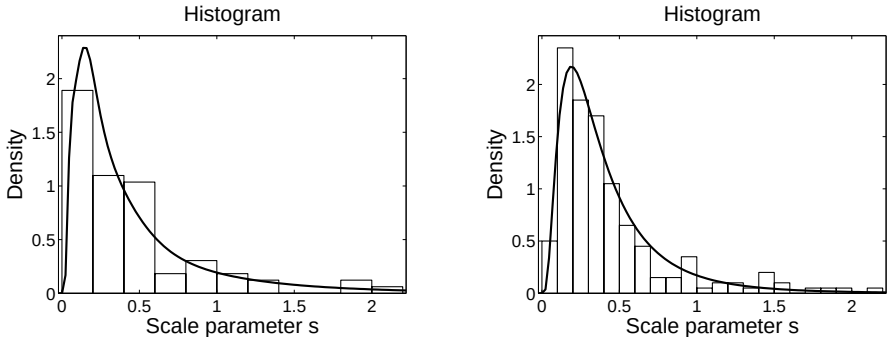
$$s = \frac{fR}{Z_0}. \tag{4.24}$$

From (4.24) we can see that the scale depends only on Z_0 which makes intuitive sense - we expect the scale of the image of an object to be approximately proportional to the reciprocal of the viewing distance with small distances from the camera producing larger imaged objects and vice versa. A similar relation between scale and distance applies to more general objects. An illustration of this relationship with additive Gaussian noise is shown in Fig. 4.7.

In order to randomly sample the scale distribution we conducted a simple experiment in which we try to simulate a typical computer vision scenario. In this experiment a rotation-invariant object (e.g. a ball) is placed inside a room and pictures of it are taken from a variety of distances and positions in the upper-half of the viewing sphere. We have chosen to use a ball as a test object because its shape does not change as the angle of the camera changes and thus we can focus on the effects of scaling alone. In addition, because perspective distortions do not have a strong influence on the ball's shape we can view the scene from nearby and so obtain a much more complete range of samples. A separate image of the ball is also taken that serves as a prototype template (see Fig. 4.8(b)). This prototype template is assumed

(a) (b)

Figure 4.8: Typical captured image sample (a) and the prototype template (b).



(a) (b)

Figure 4.9: Sampled histograms of the scale from the first (a) and second (b) experiments.

to have scale parameter $s = 1$ and is used as a reference for the sampling process. We carried out two separate experiments in which two different people were instructed to take pictures of the object placed inside a room (different rooms for each experiment) from a variety of distances and angles. We did not specify how many photos from each location each person should take or what distance from the object to favour. The only instruction was to take photos of the object approximately centred in the image. The resulting distributions should reflect the sampling of each particular individual and the properties of the room (rectangular, square, clutter and so on). For the first experiment, we captured 82 and for the second 200, 400x300 grey-scale images of the ball. In each case this should provide enough samples to determine the scale distribution. These images are then used as input to a basic template matching system that uses the prototype template (Fig. 4.8(b)) in an energy minimisation scheme to locate instances of the ball in the image. A match is located where, as an expedient to avoid the pathologies associated with the trivial solutions at $s = 0$, the normalised sum of squared errors between the prototype template and the image is minimum. The template is allowed to translate and scale.

The resulting histograms for the two experiments are illustrated in Fig. 4.9 (a) and (b) respectively together with overlaid non-parametric estimates of their pdfs calculated using a smoothing function with a Gaussian kernel. As may be seen the scale distribution is skewed to the right, constrained to be zero at $s = 0$ and has a peak around $s = 0.2 \sim 0.3$. The peak position depends of course on the choice of the prototype template and the distance from the object to the camera we originally chose for capturing the template image. A shorter distance would create a larger template and thus would move the peak of

the histogram closer to zero, whereas a longer distance would generate a smaller template and spread out the peak of the pdf. The underlying distribution appears to be the same in both cases and all that changes is the shape and location of the parameters (e.g. mean and standard deviation). The peak is at $s < 1$ because we chose to take the template image from quite close-up so as to ensure sufficient detail was visible and to avoid having to scale-up the template too much.

If very many images were collected, it would be possible to build a fine-grained non-parametric model of the distribution of scale. We didn't collect enough images for this and instead sought to ascertain which parametric model distribution would explain the data. Parametric models in general have greater efficiency at the cost of more specific assumptions about the data but it is important to verify whether the assumed distribution is indeed valid.

Our goal therefore is to find a good distribution model that best describes the scaling of objects. There is a large number distributions that might be good models for our data. However, we will restrict ourselves to consideration of the following models owing to their tractability and simplicity:

- Normal distribution with pdf:

$$N(s) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{s-\alpha}{\sigma}\right)^2\right] \tag{4.25}$$

- Weibull distribution with pdf:

$$W(s) = b\sigma^{-b}s^{b-1} \exp\left[-\left(\frac{s}{\sigma}\right)^b\right] \tag{4.26}$$

- Exponential distribution with pdf:

$$E(s) = \frac{1}{\sigma} \exp\left(\frac{\alpha - s}{\sigma}\right) \tag{4.27}$$

- The Wald distribution (inverse Gaussian), with pdf:

$$G(s) = \sqrt{\frac{\sigma}{2\pi s^3}} \exp\left[-\frac{\sigma}{2s}\left(\frac{s-b}{b}\right)^2\right] \tag{4.28}$$

- The lognormal distribution, with pdf:

$$L(s) = \frac{1}{sb\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\log(s)-\sigma}{b}\right)^2\right] \tag{4.29}$$

In the above, a, b, σ are the parameters of the distributions that determine their location, shape and width respectively. s is the variate that represents the scale of an object. Distributions (4.26), (4.28) and (4.29) are constrained to be zero at $s = 0$. Some of the distributions are positively skewed (for a specific range of parametric values) and give us the option of adjusting the location and width of the peak of their pdf.

To determine how well a specific distribution model fits our data (goodness-of-fit) we used a com-

bination of graphical techniques used in exploratory data analysis [Leinhardt and Leinhardt (1980)] and quantitative techniques from classical statistics. Details of these methods are described in detail in Appendix B.

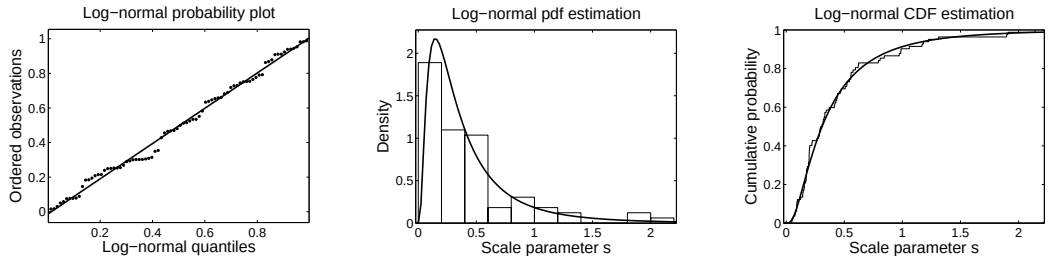
For economy of space we only show here the results from the tests on the lognormal distribution model and on the first dataset. This does not affect the generality of our assumptions since the test results are similar for both datasets. The full results on all models are included in a paper being prepared for publication [Zografos and Buxton (2005b)]. We begin by generating the lognormal probability plot (Fig. 4.10(a)) to assess whether or not our data follows the lognormal distribution. We see that the lognormal quantiles and our observations are on the same diagonal without any large deviations. If we additionally fit a line to the 25^{th} and 75^{th} percentiles we see that it is almost coincidental with the plot. This is a further indication that the data is lognormal. We can also see that the estimated pdf (via maximum likelihood) closely resembles the data histogram (Fig. 4.10(b)) and that the empirical cdf and the fitted cdf are almost identical (Fig. 4.10(c)). In the same figure we also show the residual errors from the line fitting to the lognormal probability plot: the sum of squared errors (SSE) and root mean squared error (RMSE). The closer they are to zero the better the fit. The values given are amongst the smallest values obtained from all the models we tested. In the same table we have included the R^2 metric adjusted for the residual degrees of freedom. It is defined as:

$$\text{adjusted-}R^2 = 1 - \frac{\text{SSE}(N-1)}{\text{SST}(u)} \tag{4.30}$$

where SSE is the sum of squared errors, SST is the sum of squared errors about the mean, and $u = N - m$ the degrees of freedom with N being the number of samples and m being the number of fitted coefficients estimated. The adjusted-R^2 explains the total variation in the data about the mean with a value closer to the maximum of 1 indicating a better fit. In this example we see that the line fitted explains about 99% of the data variation which indicates that the data in the probability plot is almost perfectly linear.

Our quantitative analysis results together with the maximum likelihood estimates are shown in Table 4.1. We can see that both the K-S and A-D tests accept the null hypothesis H_0 that the sample has come from a lognormal distribution. The high p-value additionally indicates that the results are not statistically significant at the 5% significance level. Note also that the K-S and A-D statistics are considerably lower than their respective critical values at the same level. All these results demonstrate that both tests were very much inside the acceptance region defined by the critical values. We may therefore conclude we have sufficient evidence to accept H_0 in this case. From the above and the results in [Zografos and Buxton (2005b)] we may claim that the lognormal distribution is appropriate for describing the scaling of objects in computer vision applications.

In addition to the two experiments described above we carried out a third experiment whereby we used a similar setting (spherical object placed inside a room) but in this case we generated a video sequence that simulates a person walking inside the room and looking at the object. In this way we tried to generate samples from a more realistic, natural object recognition situation. Our aim was to determine if the lognormal distribution is still a good model to describe the scale sampling process under this video



(a) (b) (c)

SSE	0.0537
Adjusted R^2	0.9922
RMSE	0.0259

Figure 4.10: Lognormal probability plot (a), estimated pdf (b) and cdf (c) plots from sample data.

scenario. The experiment involved a person holding a video camera entering the room and looking at the sphere. The sphere remained approximately in the centre of the camera view while the person was randomly walking around the room. In total, we generated approximately 90 seconds of video (2175 frames at 378×288 pixels) and then sampled one frame in every 15 to generate a total of 145 input images. The scale parameter was then determined in the same way as in the previous two experiments.

By carrying out a similar analysis to that described above for the image snap-shots, we obtained the results shown in Fig. 4.11. Here we can see that as in the earlier examples the lognormal distribution provides a good fit to our data set and further reinforces our assumption that the scale parameter (under a typical viewing environment) is drawn from a lognormal distribution. There is however one important point we should mention for this dataset. Because of the way the data samples are generated (using a video camera and walking around the room as opposed to "jumping" to random places and taking photographs) there is a strong dependence between one video frame and the next (i.e. it is possible approximately to predict the position and scale of the sphere in the next frame) even between every 15th frame which is our sampling frequency. See Fig. 4.11(d) for the high sample autocorrelation levels. This means that we cannot generate samples drawn randomly from the scale distribution by randomly moving around in the room. Some of our statistical tests that depend on this randomness criterion will thus in principle not be valid.

4.5 Experimental results

In this section we present some basic experiments carried out on our 2-D object recognition method using the objective function in (4.19). We carried out a limited number of tests on grey-scale, real images (such as the ones in Fig. 4.12 (a) and (d)) as a proof-of-concept study rather than an exhaustive evaluation of our method. As we mentioned earlier, the 2-D solution is but an initial investigative step on the way to developing the 3-D object recognition method and so extensive tests are not required. For the 3-D case, however, which is the main focus of this thesis we have carried out a number of more detailed experiments and analysis.

In both the illustrated cases, the template (Fig. 4.12(b) and superimposed rectangle in Fig. 4.12(d))

Maximum Likelihood			
Shape (b)	0.86307	std. error	0.06781
Log-Scale (σ)	-1.1757	std. error	0.09531
95% confidence interval for shape	0.74819	1.01996	
95% confidence interval for log-scale	-1.3654	-0.9861	
Kolmogorov-Smirnov			
p-value	0.5355		
K-S statistic	0.0877		
Cutoff value	0.1478		
Hypothesis at 5% interval	Accept		
Anderson-Darling			
A-D statistic	0.3394	adjusted	0.3547
Critical value at 95%	0.754		
Hypothesis at 5% interval	Accept		

Table 4.1: Quantitative results for the lognormal distribution.

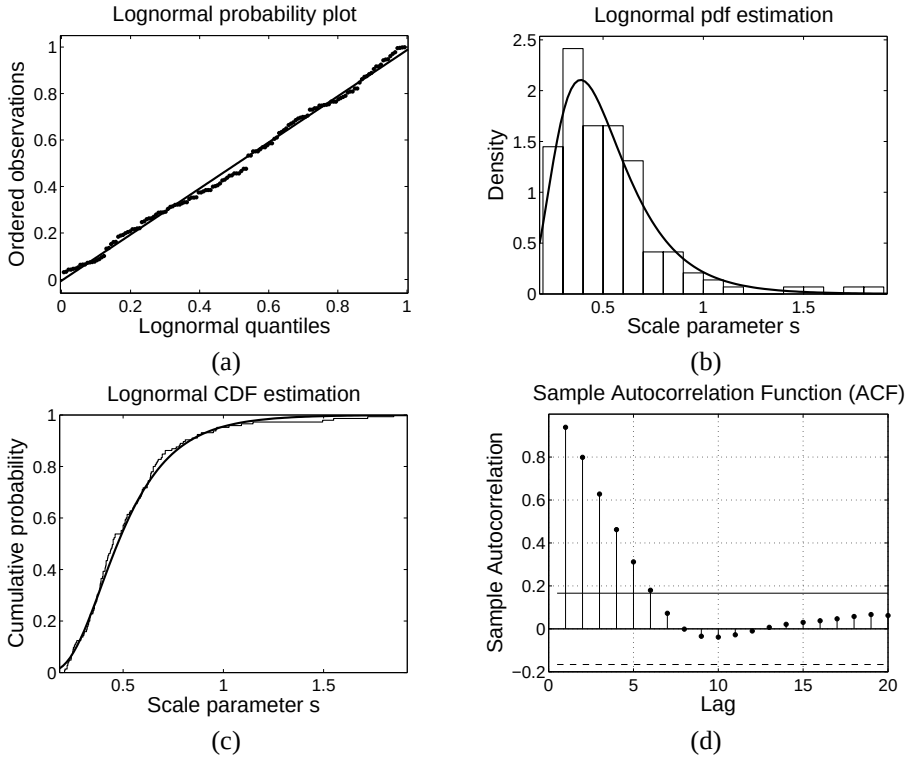


Figure 4.11: Video sequence results. (a) prob. plot, (b) pdf plot, (c) cdf plot and (d) lag plot.

is taken directly from the image (which implies the same lighting conditions) and is subjected to a random affine, geometric transformation. During matching we aim to recover (or get as close as possible to recovering) the parameters of this transformation. The first example (Fig. 4.12(a)) compares the effects of using the SSD metric without any prior information to the use of the smooth Huber metric with the combined prior distributions we have seen previously In both cases we have run 10 tests with the same optimisation algorithm (differential evolution [Storn and Price (1997)] and to be discussed in Chapter 6) and under similar settings. For the first case where there is no prior we have manually to restrict the optimisation algorithm away from the trivial solutions at $s = 0$. We do so by assigning an infinitely large error value to any solution of $s < 0.5$ (see Fig. 4.13(a)). To illustrate just how much better an approach based on a Huber metric combined with a probabilistic prior is we present in Fig. 4.13(b) the Euclidean distances of the recovered coefficient values from the known, ground truth solution for all the 10 test runs and for both cases. It is immediately obvious that the Huber & prior combination outperforms the SSD-only approach in recovering solutions closer to the ground truth in every test case.

This is also illustrated in the second set of tests in the images in Fig. 4.12(d) where the average of 10 tests runs using the Huber & prior combination are displayed in Table 4.2 Here we see just how close the optimisation algorithm has managed to get to the actual solution. A typical good, identified result for both images can be seen in Fig. 4.12 (d) and (e).

Furthermore, we show the effects of using both the lognormal and Gaussian priors on the log-posterior probability. In this example we have isolated the scale space by choosing a rectangular template (e.g. the female face in Fig. 4.12(b)) and varying the scale parameter s while keeping all other parameters constant at their optimal values. The result is the log-likelihood plot in Fig. 4.14(a). The non-trivial value of s that minimises the residual error is correctly $s = 1$ and we note that for $s > 1$ the error grows parabolically. However, we also note that for $s < 0.5$ the error becomes very small and eventually drops to zero for $s = 0$. This clearly does not constitute a meaningful answer but is a case of a trivial solution we mentioned previously. If we initialise an optimisation algorithm close to $s = 0.5$ it might converge to the trivial solution $s = 0$ which in the presence of noise will be lower than the desired solution at $s = 1$ and might thus cause global optimisation algorithms to produce the wrong results. Unfortunately, we cannot know beforehand which values to use as constraints in our optimisation algorithm (i.e. $s <> 0.5$) since the critical value is not fixed but varies in relation to the true optimal value s as determined by the size of the template used. We also note as discussed in [Buxton and Zografos (2005)] that the problem should not occur if the background is included in the modelling process. In that case when the template shrinks to zero the foreground object of interest will not then match the assumed background.

If we now use a lognormal prior (Fig.4.14(b)) that is fairly platykurtic we get the resulting log-posterior distribution (Fig.4.14(c)). The problem with the trivial solution has been rectified by assigning a very low probability (or a very large inverse log-probability) for scale parameter values close to zero and the objective function has been regularised so that it has one global minimum that is the correct solution. This can easily be located with common deterministic, local optimisation algorithms.

In the same example we show the use of a Gaussian prior (Fig.4.14(c), dashed line) with parameters

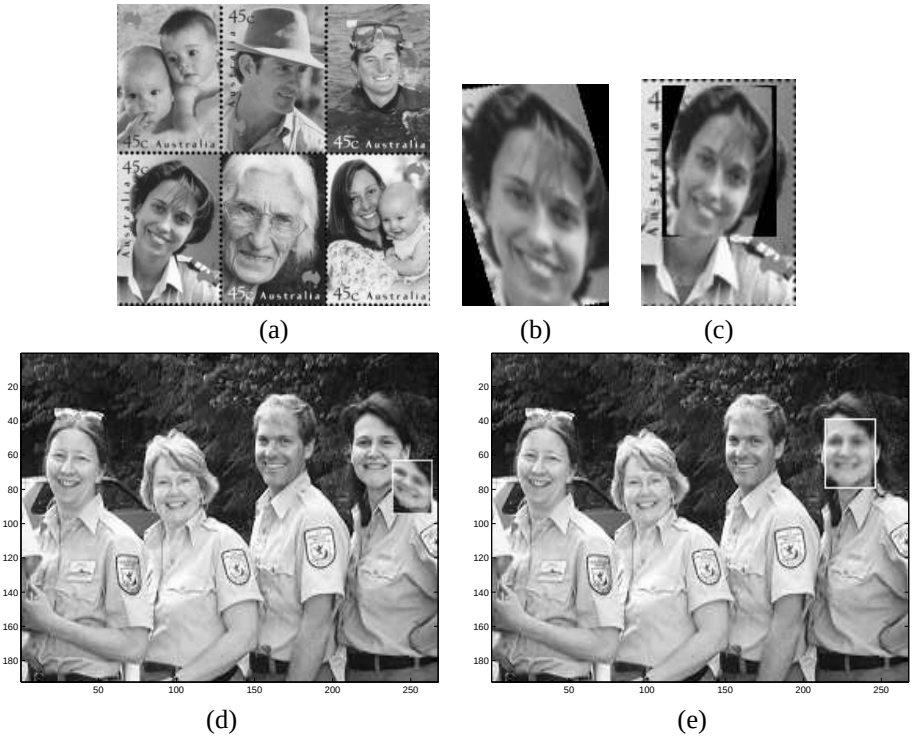


Figure 4.12: Experiments on real images with randomly transformed templates.

$\sigma = 2$ and $\mu = 1.25$ chosen in order for the pdf approximately to have high probability around the same range of values as for the lognormal prior. The resulting posterior distribution (Fig. 4.14(c), dashed line) shows that the regularisation effects for values $s < 1$ are not as strong as in the case of the lognormal prior and it creates a flat objective function with the desired minimum at $s = 1$ more difficult to find. If we decrease the standard deviation σ the situation somewhat improves with the objective function for $s < 1$ becoming steeper but this overly biases the posterior and may not be desirable in most cases. If on the other hand we increase σ the posterior for $s < 1$ becomes flatter until σ is increased so much that the Gaussian prior tends to become a uniform distribution which as we know does not have any effect on the likelihood. Perhaps the only advantage in using a Gaussian prior is that the tuning of its parameters corresponds to more intuitive changes in the shape of the pdf than for the lognormal prior.

The effects of a lognormal prior in two dimensions are also shown in Fig. 4.14(d), (e) and (f). As we can see in this case, the above problems are exacerbated with a very narrow basin of attraction (4.14(d)) and the existence of an infinite number of trivial solutions for when s_x and s_y are close to zero. Using a lognormal prior (Fig. 4.14(d)) can dramatically improve the situation by creating a convex error surface with a single global minimum (Fig. 4.14(e)).

Since the sum of squares likelihood for any image will exhibit this typical behaviour[5] we may say that in general the lognormal produces more desirable regularisation results than other commonly used priors without unnecessarily biasing the posterior.

[5]Unless of course we normalise by the size of the template. Although this will solve the problem of trivial solutions it will introduce unwanted noise and thus many local minima in the objective function for s close to zero.

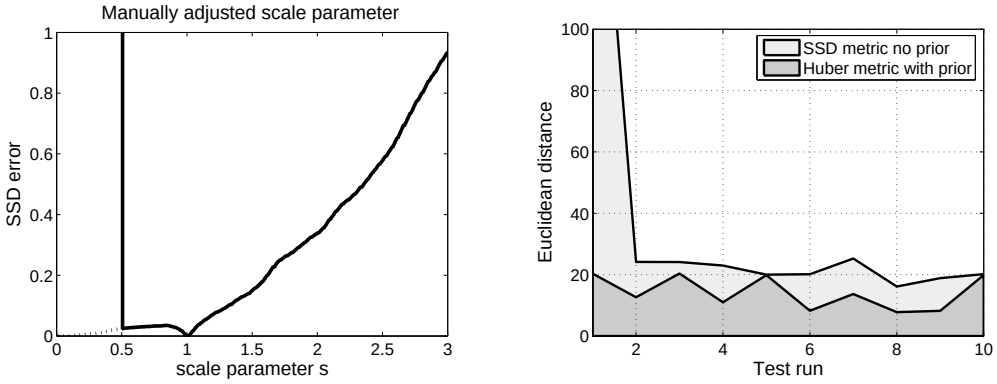


Figure 4.13: (a) Manually adjusted scale space and (b) comparison between Euclidean distances.

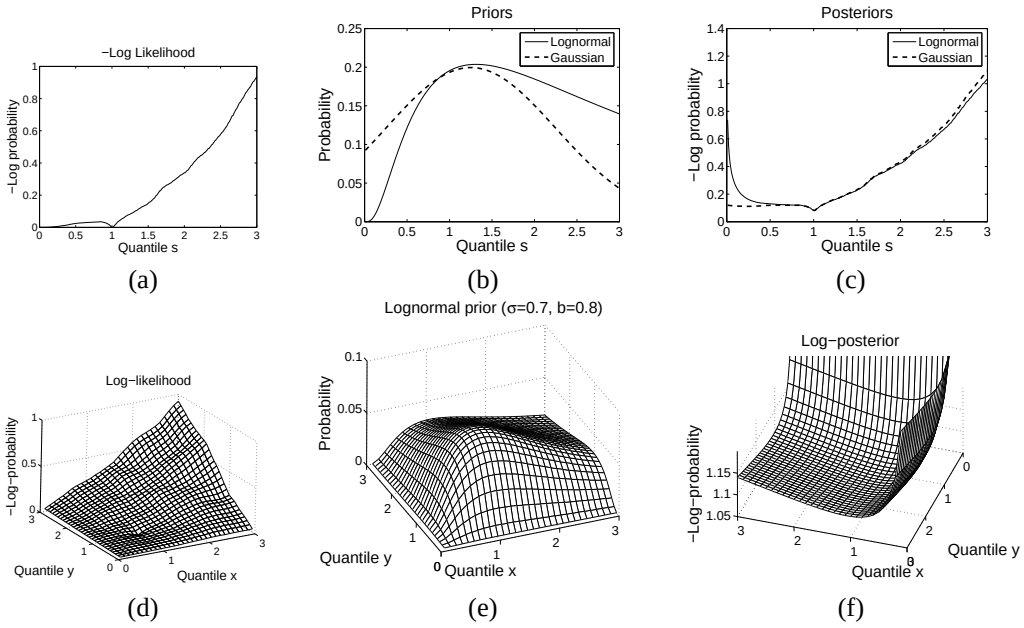


Figure 4.14: The effects of the lognormal prior on the scale parameter surface.

Transformation	Actual	Estimated	Absolute deviation
Rotation (ϑ)	30.47^o	29.7046^o	0.7654^o
Translation (d_x, d_y)	$211, 37$	$213, 38$	$2, 1$
Scale (s_x, s_y)	$1.3077, 1.1923$	$1.3125, 1.2719$	$0.0048, 0.0796$
Shear (φ)	27^o	24.6776^o	2.3224^o

Table 4.2: Comparison between actual and estimated transformation values from Fig. 4.12(d),(e).

4.6 Basic foreground/background modelling

One typical problem with template matching (seen for example in Fig. 4.1) is the fact that we may be faced with a very narrow basin of attraction in the error landscape surrounding the desired solution and also when similarity or affine geometric transformations are used with spurious, trivial solutions. We have also noted in passing that, from a probabilistic point of view, it is not correct to match the template only to the region of the target image covered by the template as this amounts to changing the data to be explained according to the hypothesised location, size and shape of the model. The data should be fixed independent of the hypothesis and it is the whole image that should be explained. It is therefore necessary, as noted earlier, to model both the object of interest and the image background and to match both to the whole image. A correctly chosen model correctly located over a foreground object in the target or scene images will thus generate only small residuals throughout the image. An incorrectly chosen template model and/or one incorrectly located will however generate large residuals both from the area under the template and from the region of the foreground object in the target image which, will not match the background model.

Furthermore, we expect such problems to be exacerbated when the transformation T of the template includes photometric transformations in addition to geometric transformations as they can allow an incorrectly located template model to adapt to some extent to the background of the target image and the background model to adapt to adapt the foreground object of interest. Similar deleterious effects will occur for an incorrectly chosen template model.

To illustrate such problems we consider the simple scenario of matching a template $I'_m(x', y')$ to a target or scene image $I_T(x, y)$ under affine photometric (grey-level) and affine geometric transformations of the kind:

$$I_m(x', y') = aI'_m(x', y') + b, \tag{4.31}$$

$$\begin{aligned} x &= a_0 + a_1 x' + a_2 y' \\ y &= b_0 + b_1 x' + b_2 y' \end{aligned} . \tag{4.32}$$

In (4.31) $I'_m(x', y')$ and $I_m(x', y')$ stand respectively for the template intensities at pixel (x', y') before and after the photometric transformation whilst in (4.32) the pixel coordinates x', y' before the geometric transformation are mapped into image coordinates (x, y). The net effect of the two transformations is to map $I'_m(x', y')$ into $I_m(x, y)$. In this example our matching criterion is a SSD error measure:

$$\min \left\{ \sum_{x,y} (I_T(x, y) - aI'_m(x, y) - b)^2 \right\}. \tag{4.33}$$

Minimisation over the parameters (a, b) of the photometric transformation may be carried out analytically and the result written in the following form:

$$\min \left\{ \left\langle {\Delta I_T}^2 \right\rangle (1 - c^2) \right\}, \tag{4.34}$$

where $\langle \ldots \rangle$ stands for an average over the pixels (x, y) in the summation, $\Delta I = I - \langle I \rangle$, and c is the correlation coefficient defined as:

$$c = \frac{\langle \Delta I \Delta I_m \rangle}{\sqrt{\langle \Delta I^2 \rangle \langle \Delta I_m^2 \rangle}}. \tag{4.35}$$

Except for the term in $< \Delta {I_T}^2 >$, (4.34) is one of the many familiar image matching criteria whose performance in template matching have been evaluated several times [Tsai et al. (2003); Brown (1992)]. Other familiar forms in which the deviations from the mean intensity are used, or the intensities normalised for the image brightness or level of illumination may similarly be derived by using the photometric transformations which respectively include only the bias b or contrast or gain a.

The result (4.34), in particular the presence of the term $\left\langle \Delta I^2 \right\rangle$ deserves closer scrutiny. First we note that the SSD is usually computed by summing over the pixels lying within the image area A_m, say, covered by the transformed template $I_m(x, y)$. If the geometric transformation (4.32) is restricted to translation of the template and if the variance $\left\langle \Delta I^2 \right\rangle$ were independent of the position of the template (4.34) would then reduce simply to maximisation of the magnitude of the correlation coefficient c. However, this will generally not be so and $\left\langle \Delta I^2 \right\rangle$ cannot be removed from (4.34) without changing the matching criterion. A number of difficulties then become apparent:

1. Bland regions of the image where there is little or no variation produce good matches with little error to *any* object by virtue of setting $a = 0$ and $b = I_T$. In particular dark regions of the target image with $I_T \sim 0$ will match to any template with little error.

2. If we retain the affine geometric transformation (4.32) the area A_m covered by the transformed template may under scaling or shearing shrink to zero resulting in a zero variance $\left\langle \Delta I^2 \right\rangle$ and spurious matches.

One way to remove such spurious matches is, as noted earlier, to normalise by the area A_m but this means that the matching score becomes very noisy whenever A_m is small. Furthermore, there seems no straightforward way of arriving at such a measure within a probabilistic approach. Another way which is straightforwardly within the probabilistic approach, is to introduce suitable priors which will add regularising terms to criterion (4.33) and bias against spurious solutions in which the template is shrunk to cover only a very small area.

Adopting the probabilistic viewpoint is very satisfying, but exposes a more fundamental failing of the approach outlined above. As we have already indicated several times, by using only the area under the transformed template in the match criterion (4.33), the observations we are using to test our hypothesis as to where the object is in the image (which may include the null hypothesis that the object of interest is not present) become dependent on the parameters of our model, i.e. on the hypothesis. As pointed out by [Sullivan et al. (1999)], this is not correct in a Bayesian approach. Simply put, our observation is the whole of the image and we should have a model of the background as well as of the foreground object or objects of interest. Thus, we should utilise not only positive evidence of where we are hypothesizing the object or objects may be, but also negative evidence from elsewhere in the image where the observed image intensity does not accord with our expectations for the background.

We should therefore include *all* pixels in the image in the sum in our SSD score (4.33). The variance $\langle \Delta I^2 \rangle$ is then evaluated over the *whole of the image area* A, say. One nice outcome of this view is that we do not have to worry about the possibility of the variance $\langle \Delta I^2 \rangle$ vanishing unless there are trivial, totally bland images in the data which can easily be detected and removed.

One downside of constructing a foreground/background model is that the combined model will necessarily be more complicated than the foreground model alone and, most probably, less applicable and therefore more fragile than a model which only includes the foreground. We thus either have to know what the background is, build a very simple model, or have a statistical model of what it is expected to be like. In fact, it is surprisingly often the case that we know the background or may learn it. Examples include: medical applications, many monitoring and some inspection systems. Indeed, in the latter, it is often an essential requirement that the background is known or has to be modelled [Zhou and Aggarwal (2001)]. In some cases, as in the CMU PIE database [Sim et al. (2002)], the background has been recorded with no objects present (in this case human faces) for the convenience of researchers.

To illustrate several of the above points we construct a very simple foreground/background modelling example. Our basic assumption is that there is an object of area A_O of constant intensity I_O in the foreground of an image $I_T(x, y)$ of area A which otherwise is of constant intensity I_B. The model correspondingly has a foreground object of intensity I_m of area A_m centred at (x_m, y_m) and a background intensity I_b. The model and object may have an overlap area A_{Om} as sketched in Fig. 4.15(a). For simplicity, given that the model contains foreground and background intensities I_m and I_b that we may vary we shall ignore the photometric transformation (4.31) and, since we have not specified the size or shape of the model of area A_m, we will similarly ignore the geometric transformations (4.32).

For our simple model calculation of the match score such as the SSD is a matter of counting the number of pixels in, or the areas of, four contributions where: the model template overlaps the image object, the model template overlaps the image background and vice-versa, and where the two backgrounds overlap. This leads to:

$$\min \left\{ \left[\begin{array}{l} (A_m - A_{Om})(I_B - I_m)^2 + A_{Om}(I_O - I_m)^2 + \\ (A_O - A_{Om})(I_O - I_b)^2 + (A - A_m - A_O + A_{Om})(I_B - I_b)^2 \end{array} \right] \right\}. \tag{4.36}$$

In (4.36) the area of the overlap A_{Om} is a function of the co-ordinates (x_m, y_m). Even for simple objects such as rectangles and circles A_{Om} is complicated and non-analytic. Optimisation over (x_m, y_m) (and in general any other model parameters determining the orientation, size, and shape of the model object, i.e. affecting A_m and A_{Om}) thus has to be carried out numerically. However, we may choose in the above whether to treat the photometric values in the model, I_m and I_b, as constants or as variables and in the latter case carry out optimisation with respect to them analytically. Thus, for a traditional rigid, windowed template, I_m would be constant and, since we only need the first two contributions in (4.36)

from under the template, I_b is irrelevant. It follows that in this case (4.36) becomes simply:

$$\min\left\{\left[(A_m - A_{Om})(I_B - I_m)^2 + A_{Om}(I_O - I_m)^2\right]\right\}, \tag{4.37}$$

in which, if we choose the foreground and background intensities correctly to match the image, $(I_B - I_m)^2$ may be replaced by $(I_B - I_O)^2$ and $(I_O - I_m)^2$ by zero. However, if the object model intensity I_m is not fixed and we optimise (4.37) with respect to it we find that (4.37) is replaced by:

$$\min\left\{\left[(A_m - A_{Om})(I_O - I_B)^2 A_{Om}/A_m\right]\right\}. \tag{4.38}$$

Whilst (4.37) has, as expected, a single basin of attraction of area $\sim 4A_O$ containing at its unique minimum the correct location of the object (see Fig.4.15(b)), (4.38) does not behave in such a nice way. There is a much smaller basin of attraction and it is surrounded by a rim beyond which there is no overlap and the matching score becomes zero as I_m adapts to the image background level (Fig.4.15(c)). This simple behaviour is symptomatic of what can happen if adaptive or flexible models are not used carefully. Somewhat surprisingly simply taking into account all the evidence from the whole of the image largely alleviates the problem. In this case, we need to optimise (4.37) with respect to both I_m and I_B which, if $A_m = A_O$, leads to:

$$\min\left\{\begin{array}{l}(I_O - I_B)^2(A_m - A_{Om}) \\ [A_{Om}/A_m + ((A - A_m) - (A_m - A_{Om}))/(A - A_m)]\end{array}\right\}. \tag{4.39}$$

This has a single basin of attraction, slightly smaller than that in the examples above with a small rim and, when there is no overlap, a plateau slightly less high than that obtained when a rigid, windowed template was used (Fig. 4.15(c)).

In the above the basin of attraction has an area of approximately $4A_O$ and the landscape outside the basin is flat (see Figure 4.15 (b)). Structure within the object and in the background will lead to considerable variation of the SSD outside the basin of attraction. Also, the area of the basin of attraction is larger in our simple model (probably considerably much larger) than we should expect in general because:

1. Perfect correlation of the pixel intensities with each other will not persist right across the object. The object may be patterned or have systematic variation across it that will reduce the strength of the correlation and may change its sign, with the result that the range of the correlations is unlikely to extend fully across the object.

2. Structure in the foreground and background will tend to decrease the size of the basin of attraction and make the rim irregular. Noise will have a similar, but unless the images are very noisy, less pronounced effect.

Smoothing the image and model will tend to increase the range of the correlations and also, probably, their strength. However, neither effect is necessarily guaranteed in the sense that we can expect

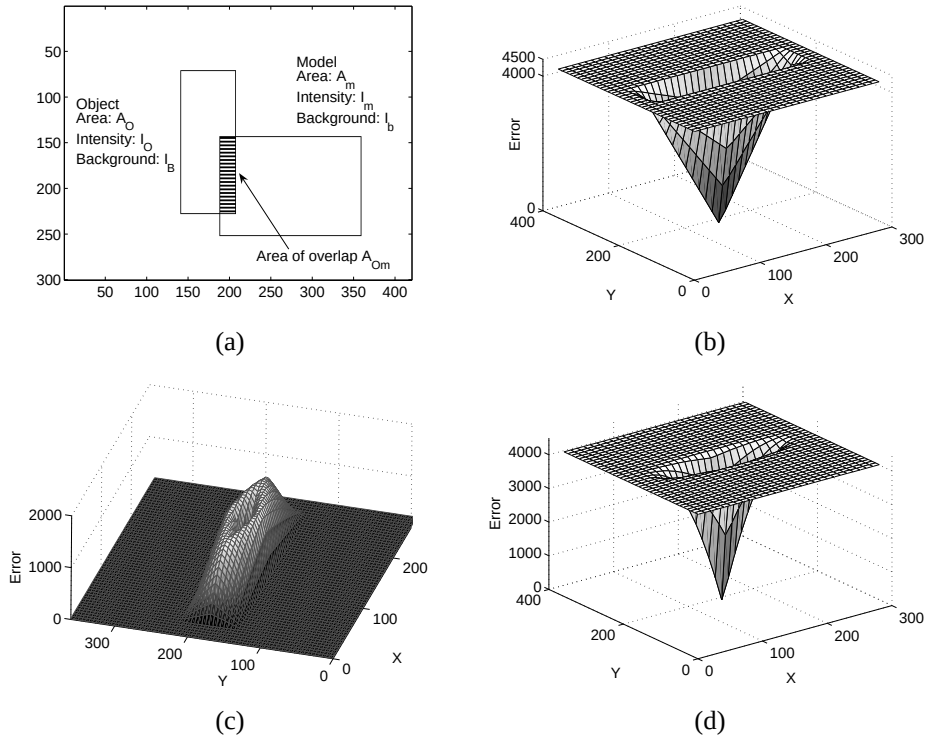


Figure 4.15: Simple matching examples and error surfaces.

such increases to occur monotonically as the smoothing is increased. In general, increased smoothing will eventually tend to wash-out distinctive features on the object and structure in the background leading to a decrease in the depth of the basin of attraction and, with enough smoothing, the merging and disappearance of some, hopefully spurious, basins of attraction.

In conclusion we may say that in template matching both foreground and background should be modelled. Doing so is necessary in order to be able to make a valid probabilistic interpretation of the matching process. In addition we can avoid at least some spurious, trivial solutions and there seems to be an improvement in the form of the error surface and localisation close to the basin of attraction. It is the case nevertheless that because of the characteristics of the matching problem the error surface will in general be rugged and of a form that renders many of the common optimisation algorithms ineffective and unreliable. This is the main reason why as we will see later we have carried out further research into evolutionary optimisation algorithms that may be able to overcome such problems.

4.7 Summary

In this chapter we have presented a robust treatment of the 2-dimensional, pixel-based, template matching approach to object recognition for intensity images using a Bayesian formulation. We distinguished between the different transformations of the template and their respective degrees of freedom and introduced individual prior distributions to restrict the deforming template to viable solutions. In addition, we examined the difficulties caused by there being different distributions of the residual errors in the matching when the template is placed in foreground and background image regions. Initially we tried to

address this problem using the Huber metric that deals with small and large error residuals, as expected respectively with the template placed in the foreground and background, in a different way. In order to gain greater insight into these and other problems, in particular concerning the probabilistic interpretation of the approach that might otherwise be overlooked in template matching, we developed a simple geometric and photometric model. This was used to explore as far as possible analytically effects caused by adaptation of the template and to explore the form of the matching objective function. Some preliminary, exploratory results for the matching of 2-D templates to real images were obtained using our method and were also presented.

Chapter 5

3-D object recognition

This chapter presents our research on 3-D object recognition and is a natural progression from the 2-dimensional case we have just examined. Since we are still working with 2-D images the same kind of theoretical framework applies here and, as a consequence, we will encounter similar problems. We present a method for model-based recognition of 3-D objects from a small number of 2-D intensity images. Our method works by using the linear combination of views (LCV) theory to combine images from two (or more) viewpoints of a 3-D object to synthesise images of novel views of the object. The object in question is recognised in a target, scene image by matching to such a synthesised novel view.

The key element in our approach is the recovery of the linear combination of views parameters. Since we are working directly with pixel intensities we suggest searching the parameter space using a powerful optimisation algorithm in order efficiently to recover the optimal parameter configuration and recognise the object in the scene.

As in the 2-D case previously discussed searching a large parameter space especially one that is very noisy and with a large number of local optima can be an arduous task even for sophisticated, modern optimisation algorithms. For this reason and continuing the theme from our earlier work, we decided to condition the error surface by incorporating probability distributions for the individual transformation parameters and build a Bayesian framework. This will allow us to create a more favourable surface with a wider basin of attraction and convex-like properties and with a well-defined global optimum; properties that should significantly aid the optimisation process.

5.1 The recognition system: Rigid objects

The recognition system we are going to present here is fairly straightforward and makes use of a number of concepts we have seen previously. It essentially has three distinct parts. First, a *modelling* part which in the work carried out for this thesis is the task that requires the most input from the user, but since it is performed off-line it does not affect the execution speed of the recognition. Second, a *synthesis* part in which a novel image is synthesised using the LCV theory (section 3.3) to calculate its geometry and intensity. Third, the *matching* part in which, with the assistance of an optimisation algorithm, we try to find the best match and determine if the object is in the scene and if it is, recover its configuration. The outline of the system is illustrated in Fig. 5.

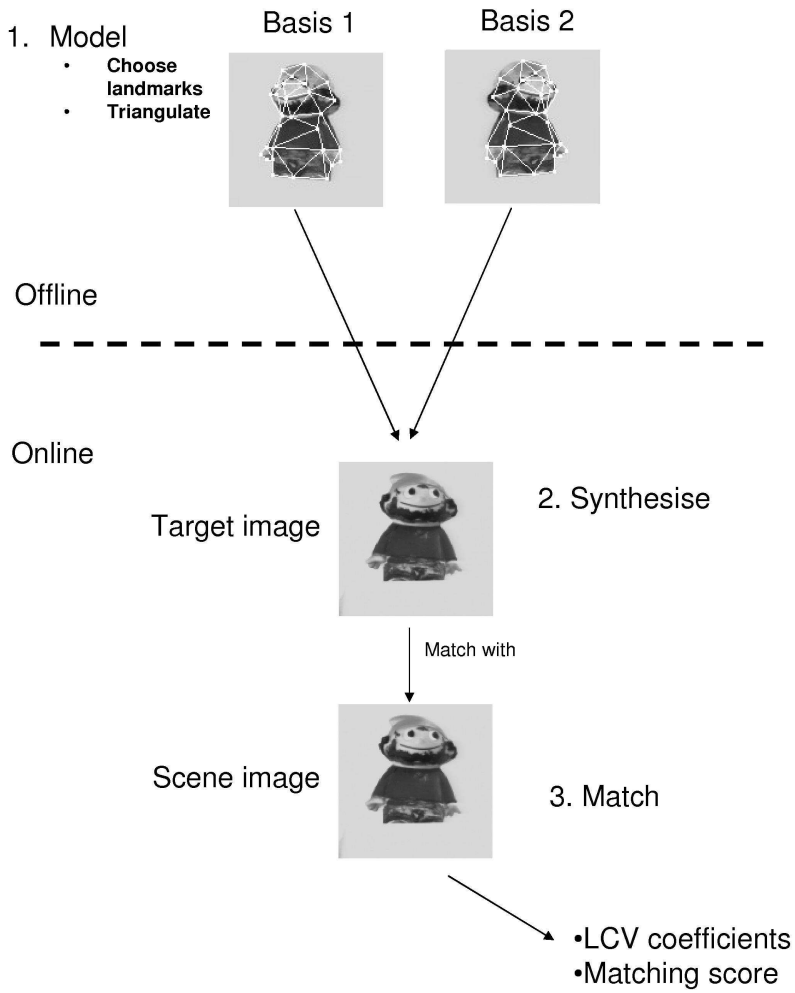


Figure 5.1: An outline of the proposed recognition system.

5.1.1 Modelling

The first stage of our approach involves the creation of a linear combination of views model for a 3-dimensional object which can be used to synthesize the novel view for matching. This requires the selection of a number of appropriate 2-D images (the basis views) that represent the object of interest as seen from different, but nearby viewpoints. As we have seen earlier in Section 3.3, we can synthesise the geometry of an affine image from a suitable selection of basis views and a set of linear coefficients. This synthesis requires the existence of a number of corresponding points (landmarks) in all the basis views and the view to be synthesized. Given such landmarks, a set of optimal LCV coefficients may be obtained by solution of a linear system of equations.

Ideally, we would like the basis views to include all the geometric and photometric detail that can be seen on the object in the scene image, without any missing or occluded regions, and with as little difference from the scene view as possible (e.g. viewed from the same or nearby aspects). If we know or can predict what the scene image will look like, or preferably the range of extrinsic variation that an object might exhibit in a given experimental setup, then manual selection of the basis views should be a straightforward task.

It is often the case however that we are only given a large set of training images of a 3-D object, captured from a variety of viewpoints across the view-sphere, without any explicit information about the scene properties. Under such conditions, manually choosing the best images to represent the basis views might be a difficult task, given the large number of possible candidates and that we do not have a quantitative measure of what might constitute a "good" set of basis images, but only the qualitative requirements stated above.

Although automatic choice of the basis views and of the model-building element is outside the scope of this thesis, we will briefly nevertheless attempt to define a numerical criterion with which to quantify the representative power of a given set of basis views. Ideally such a measure would quantify their ability to best synthesise novel views for which we take as a proxy their ability to reproduce a given set of (training) images on average. A good choice for such a measure is the root mean square error between the images synthesised from a particular pair of basis views to reproduce every other image in the training set. This in a sense measures how well a given selection of basis views can represent via the LCV synthesis a set of 2-D images. Thus, if we assume a set of n training images with landmark points $X = \{X_1, X_2, \ldots, X_n\}$ and a pair of basis views with land mark points $\{X_i, X_j\} \in X$ for $i, j = 1 \ldots n$ with $i \neq j$, we can compute the r.m.s. error:

$$\varepsilon_{\text{RMS}_{i,j}} = \sqrt{\frac{1}{n}\sum_{k=1}^{n} \varepsilon_{i,j}^2(k)}, \tag{5.1}$$

where $\varepsilon_{i,j}^2(k)$ is the squared error between an image X_k and its synthesised match produced by basis view pair $\{X_i, X_j\}$. This error is defined as:

$$\varepsilon_{i,j}^2(k) = \|X_k - C_{i,j}(k)B_{i,j}\|^2, \tag{5.2}$$

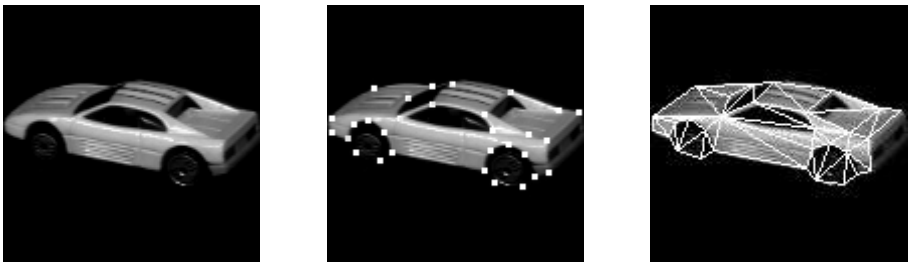
and can be thought of as the geometric difference (Euclidean distance) between the landmark point coordinates in X_k and those in their synthesised counterpart, produced by $\{X_i, X_j\}$. $C_{i,j}(k)$ is a 2×5 matrix of the LCV coefficients (see eq. 3.14) and $B_{i,j} = \left[1, X_i^T, X_j^T\right]^T$. The pair of images (i, j) that produce the lowest ε_{RMS} error is to be chosen as the basis pair. This selection step, although likely to be computationally and experimentally time consuming, would only be carried out once during off-line training. Although errors in the synthesis of the landmark points will obviously affect the appearance of the computed images they do not measure directly the accuracy with which the target images are to be reproduced. If we require a more direct measure of this than using the geometric difference between the landmark points, we can replace (5.2) with:

$$\varepsilon_{i,j}^2(k) = \left\| I(X_k) - I_{i,j}(X_k) \right\|^2, \tag{5.3}$$

where both the images $I(X_k)$ and the synthesised $I_{i,j}(X_k)$ are represented as intensity bitmaps and not as a collection of landmark points. In this way we incorporate the additional representative power of all the image pixels to improve on the selection of the most appropriate basis views.

However, as assumed in the above and as implied by (3.14), in order to recover the optimal LCV coefficients and synthesise the target, scene image I_T it is necessary to have corresponding landmark points already in I_T, meaning that we can only synthesise a known, given view. This has been shown to be very successful in particular by [Hansard and Buxton (2000b)] and suggests that the LCV approach could be useful for object recognition though in an object recognition task such landmark points will not be available a priori. Whilst, in principle, one could imagine using feature detectors to extract the required landmark points we have argued that this is unlikely to be successful and that one should proceed without any prior landmarks in or any knowledge of the geometry of the objects in the *target* image view, and instead directly search the LCV coefficient space. We do however require a sparse set of corresponding landmark points in all the *basis* views. These points are manually chosen, once, during off-line model building, to correspond with each other. Even though it might at first seem that the location of the landmark points is not very important, in practice when a modest number of landmark points is used the synthesis of the image appearance is greatly improved if landmarks are chosen to fall on to image features. This is especially the case if, as we shall see later on, the edges of the triangles defined with the landmark points as vertices should coincide with depth discontinuity boundaries [Hansard and Buxton (2000b)]. Such edges are often where strong features are located. An illustration of such landmark points can be seen in Fig. 5.2. If the landmarks are chosen as illustrated to coincide with salient points in the images only a sparse set is needed to describe objects with moderately complex geometry. Note also that we need a larger number of landmarks in areas of high curvature such as along the boundaries of smooth objects. Finally, we observe that the geometry of the object is preserved in the triangular mesh.

Manual choice of the landmark points can be a tedious and time-consuming process, especially for inexperienced users. Nevertheless, it has the distinct advantage that no outliers will arise from the selection process and that there will be no correspondence errors in the chosen landmark sets. It is expected that we will introduce some positional errors during selection of the points but because they



(a) (b) (c)

Figure 5.2: Modelling steps: (a) basis view (b) landmark points and (c) triangulation

will be small and consistently distributed errors they can easily be taken into account.

A final part of the modelling stage is the generation of consistent and corresponding triangular meshes in all the basis views. This is achieved using Delaunay triangulation [Delaunay (1934)]. It is performed in order to facilitate the computation of intensities via the representation of image regions by means of the existing landmark points, without the need for additional, dense correspondences. A triangulation is carried out only once during modelling with the same mesh topology used for all basis views. Since the landmark points are in correspondence with each other this ensures that the meshes are themselves correspondingly consistent. Furthermore, in order to preserve in the generated mesh identified strong edge structure on the object, we compute the constrained Delaunay [Shewchuk (2002)] mesh by forcing triangle edges to coincide with the locations of such boundaries. This allows us to represent the structure of non-convex objects (unlike the standard triangulation) and to separate object regions from background areas.

In conclusion, an LCV model is composed of a number of basis views representing the object of interest, a set of landmark points selected across salient points and discontinuity boundaries on these views, and a consistent triangular mesh that follows the structure of the object. All these steps may be carried out during the off-line training stage and thus do not incur any additional computational cost in the recognition process.

5.1.2 Image synthesis

To synthesise a single, target image using the LCV theory and the basis views (two in this case) we first need to determine its geometry from the landmark points. In principle, we can do so by using (3.14) and n corresponding landmark points (where $n \geq 5$) and solving the resulting system of linear equations in a least squares sense. This is straightforward if we know, can detect, or predict the landmark points in the target image I_T. Such methods may therefore be useful for image coding and for synthesis of target views of a known object [Koufakis and Buxton (1998b); Hansard and Buxton (2000b)]. For pixel-based object recognition in which we wish to avoid feature detection a direct solution is not possible but we instead use a powerful optimisation algorithm to search and recover the LCV coefficients for the synthesis. Given therefore the geometry of the target image I_T in a pixel-based approach we need to synthesise (render) its appearance (colour, texture and so on) in terms of the basis images ${I_m}'$ and ${I_m}''$. If we assume a set of landmark points have been chosen in the modelling stage we can, to a good approximation, synthesise a target image I_T as described in [Buxton et al. (1998)] from a weighted

combination:

$$I_T(x,y) = w' I_m{}'(x',y') + w'' I_m{}''(x'',y'') + \varepsilon(x,y) = I_S(x,y) + \varepsilon(x,y), \tag{5.4}$$

in which the weights w' and w'' may be calculated from the LCV coefficients to form the synthesised image I_S as we shall discuss below. Essentially, this relies on the fact that, in addition to the multi-view image geometry being to a good approximation affine, the photometry is to a good approximation affine or linear [Shashua (1992)]. (5.4) warps and blends images $I_m{}'$ and $I_m{}''$ to produce I_S. It is important to note therefore that (5.4) applies to all points (pixels) (x,y), (x',y') and (x'',y'') in images I_S, $I_m{}'$ and $I_m{}''$ and that all such triples of points are assumed to be in correspondence. Without such a dense correspondence it is not possible to use the LCV equations to map the basis views into the target image. Furthermore, in synthesizing I_S we do not require a mapping from the basis views to the co-ordinates (x,y), but the inverse mapping from (x,y) to (x',y') and (x'',y''). Since the forward LCV mapping from (x',y') and (x'',y'') to (x,y) is many-to-one this inverse is ill-posed and not defined except at the landmark points. To make the inverse well defined at all points we use the triangular mesh that was generated during the modelling stage to define a local affine transform from each triangle in the target, scene image to the corresponding triangles in each of the basis views. In other words, the image transformations from each basis to target (and vice versa) is piecewise affine and piecewise invertible. The parameters of each affine transformation can be used to map the interior (intensity) of each triangle together with its vertices (geometry) and define a dense correspondence of all the pixels between the two basis views and the target image without additional selection of landmarks. This series of piecewise linear mappings are implemented using the method of [Goshtasby (1986)]. In this way, the mapping is exact at the positions of each control-point and if the landmarks span flat (colour constant) regions of the object then the mapping is also consistent with the affine camera-model inside each triangle.

In [Koufakis and Buxton (1998b)] the weights w' and w'' were defined according to the following arguments. If in (5.4) the target I_T should coincide with either $I_m{}'$ or $I_m{}''$ then the other basis view should not contribute at all to the synthesis of I_S. We therefore have the additional implicit requirements on (5.4):

$$\begin{array}{l} \text{if } I_T = I_m{}' \text{ then } w' = 1,\ w'' = 0 \\ \text{if } I_T = I_m{}'' \text{ then } w' = 0,\ w'' = 1 \end{array}. \tag{5.5}$$

According to [Koufakis and Buxton (1998b)) and Buxton et al. (1998)] we can compute weights w', w'' consistent with the constraints in (5.5) as follows. First, we calculate the distances of the target image from each of the basis views:

$$\begin{array}{l} d'^2 = a_3^2 + a_4^2 + b_3^2 + b_4^2 \\ d''^2 = a_1^2 + a_2^2 + b_1^2 + b_2^2 \end{array}, \tag{5.6}$$

by summing and squaring the appropriate LCV coefficients. We then calculate the weights as:

$$w' = \frac{d''^2}{d'^2 + d''^2},\ \ w'' = \frac{d'^2}{d'^2 + d''^2}. \tag{5.7}$$

We can now substitute (5.7) into (5.4) and compute the geometry and intensity of the target image. The same idea may be extended to colour images by treating each spectral band as a luminance component (e.g. I_R, I_G, I_B).

5.1.3 Matching

Once a new image is synthesised from a set of linear coefficients (a_i, b_j) we need to determine how well it matches with the target, scene view. As in the 2-D case previously we employ a template matching approach using a similarity or dissimilarity metric between I_S and I_T. The comparison is carried out directly on the pixel values without any assumptions about the geometry of the scene view I_T since we do not extract features from I_T at any time during the training or matching stages.

If the match (or mismatch) score is above (or respectively below) a given threshold then the object is said to be present in the scene and its parameters are encoded in the coefficients (a_i, b_j). If desired, we may go some way to interpreting these coefficients in terms of more familiar model pose parameters, something which we will discuss later on. If the match or mismatch score does not meet the pre-determined threshold, we can generate new sets of LCV parameters, synthesise new images (i.e. object in new configurations in the scene) and check to see if we can find a better match. A suitable optimisation algorithm is used efficiently and effectively to search the large parameter space. If at the end of the optimisation the match or mismatch score still fails to meet the required threshold, then we can assume that either there exists no such object in the scene (or at least as seen from a viewpoint where it can be modelled by the LCV technique) or that the optimisation algorithm has failed to converged to a non-optimal solution. We can try to prevent the latter from occurring, at least to some extent, by using a Bayesian approach to bias the solution away from local optima, something that we will explain in the next section.

Before turning to the Bayesian approach, we recall that in order to make a valid probabilistic interpretation of the match one must compare the pixels in both the foreground and background, such as in [Sullivan et al. (2001)]. As discussed previously, the background must therefore be known (e.g. as in the CMU PIE database [Sim et al. (2002)]), or very simple (e.g. a uniform, black background as in the COIL-20 database [Nene et al. (1996)]) or itself calculated from an appropriate model. Making the comparison over all pixels in this way means that either a similarity or dissimilarity metric may be used without generating spurious solutions, for example, when the area of the foreground region covered by the object shrinks to zero [Buxton and Zografos (2005)]. We saw the problems caused by such trivial solutions in our preliminary on 2-D object recognition in the previous chapter. Within the context of the recognition of 3-D objects via our LCV approach, the possibility of such spurious solutions could, given a high dimensional parameter space, be even more damaging.

Optimisation

The recovery of the LCV coefficients requires the search of a high-dimensional space for all the possible transformations between the model and the scene. Our objective is to find the optimal model configuration that will bring the synthesised and scene images into agreement. Such a search of the 10-dimensional LCV space is computationally expensive and so we need to use an efficient method for the recovery of

the optimal coefficient set.

For this purpose we have considered the use of various global, numerical optimisation algorithms as the final stage of our object recognition system. The aim is find an algorithm that is efficient and use of which is therefore computationally feasible yet will converge to the optimum solution from a remote position in the transformation space. The examination of such methods and their combination with local optimisation techniques for improving the efficiency of the search in its final stages is the main focus of the next chapter.

5.1.4 Coefficient variation

Before we describe the Bayesian model, we would like to remark that since the pose information is implicitly encoded in the 10 coefficients (a_i, b_j), it is useful to investigate their variation as the object's pose changes in relation to the viewing direction. We are particularly interested in what we refer to as a "horizontal rotation" of the viewpoint around the portion of the view sphere defined between the two basis views. This nomenclature reflects the set-up for the simple experiment we have devised to try and recover some information about the range, the distribution and the variation of the LCV coefficients as an object is allowed to rotate between views that generate images $I_m{}'$ and $I_m{}''$.

In brief the experiment is as follows. We have used a synthetic 3-D model of a human head over a black background (Fig. 5.3 (a)) and selected a number of landmarks on prominent features of the face and along main discontinuity boundaries. To avoid introducing any manual error the landmarks were chosen from amongst the set of model vertices. The 3-D model was then allowed to rotate about a vertical axis between $\pm 20^o$ from the frontal position, and 2-D snapshots of the scene were taken under orthographic projection at 1^o intervals. The two images at $\pm 20^o$ of rotation where chosen as the basis views so all the synthesized images would be interpolated between the basis views. Since we worked directly with a 3-D model the positions of the vertices and thus the landmarks were always known within a high degree of precision.

We proceeded to evaluate the coefficients (a_i, b_j) by solving the linear system in (3.14) at each interval of rotation and thus obtained a set of coefficients for the pure, isolated, horizontal rotation between the two basis views dependent only on the rotation angle ϑ. This information enables us to draw certain conclusions about the properties of the coefficients (a_i, b_j). First we plot the graphs illustrating the variation of the 10 coefficients according to the angle ϑ. Recall that the a_i coefficients describe the horizontal x-coordinates of the target image while b_j describe the vertical y-coordinates and that a_0, b_0 are the constant terms that represent the translation between the target and basis views. For that reason, a priori we would expect a large range of possible values for the coefficients a_0, b_0. However, specifically for the rotation described, we expect only the translation along the x-axis (represented by a_0) to vary over a significantly large range while that on the y-axis should be small and show little variation ($b_0 \sim$ zero). As we can see from the graphs (Fig. 5.3) the variation of the coefficient a_0 follows a quadratic curve, coefficients a_1 and a_3 a linear curve and the remaining coefficients are constant. We note also that a_1 and a_3 have a range of $[0, 1]$ with $a_1, a_3 = 0.5$ for $\vartheta = 0^o$ (frontal view). Likewise, for the frontal view, a_0 is at a minimum. Finally, we observe that, $a_2, a_4, b_0, b_1, b_3 = 0$ and $b_2, b_4 = 0.5$.

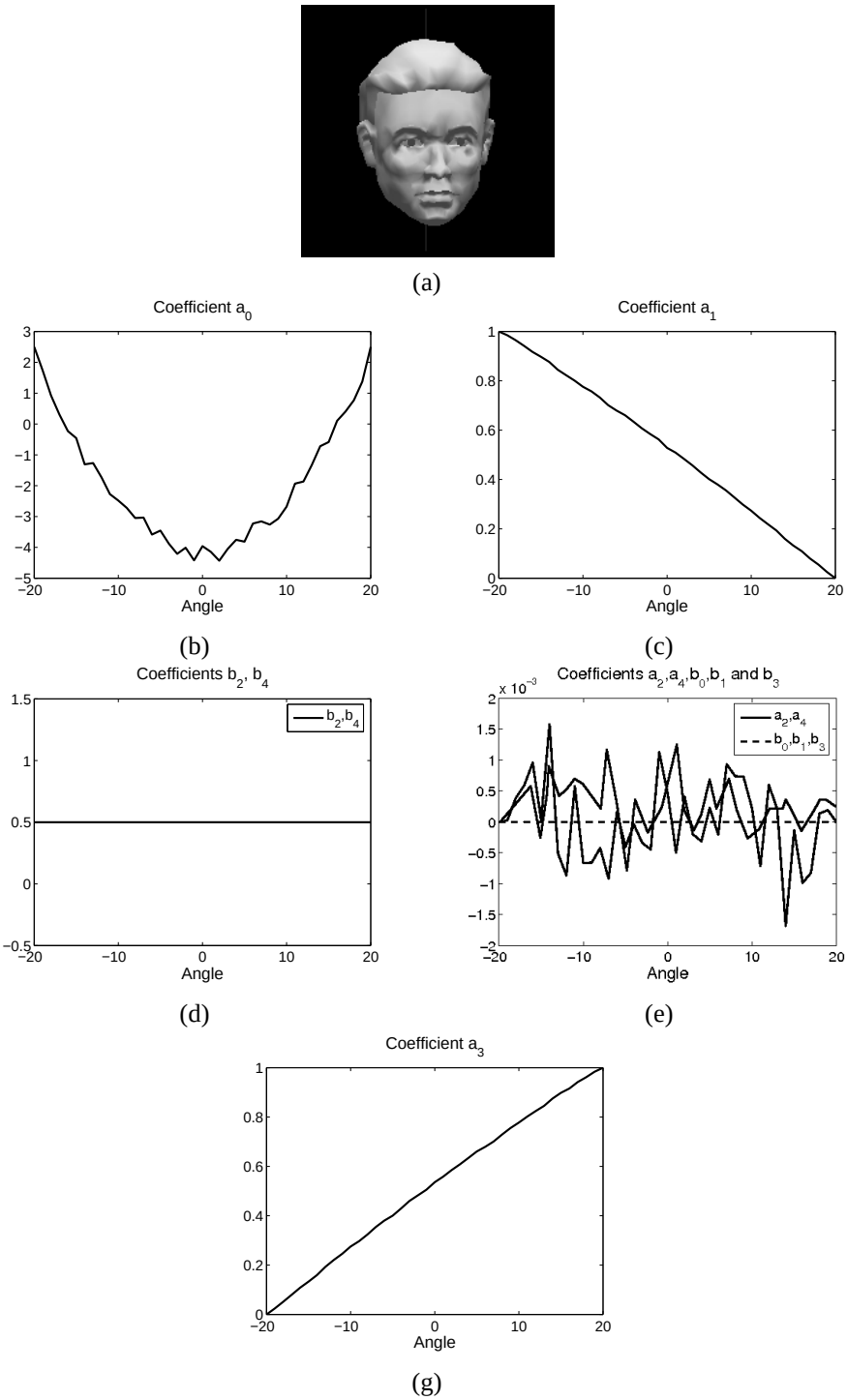


Figure 5.3: The variation of the 10 coefficients for horizontal rotation.

This information on the range of values taken by each of the coefficients can be used as "hard boundaries" or even to provide approximately regions within which we can initialise the optimisation search. Additionally, since we can determine the coefficients as a function of ϑ, we can predict the approximate solution set (always accounting for some degree of error) at each ϑ between ${I_m}'$ and ${I_m}''$. This approximate information combined with knowledge on the range of poses we are likely to encounter in a specific experiment can be used to set the means, which usually are in regions of high probability, and the widths of the Bayesian priors so as to facilitate the optimisation process. This is described in more detail in the next section. One last piece of information that may be inferred from the above experiment which, although is not employed in this work could be used during optimisation, is the distribution of each of the coefficients (a_i, b_j). For example, we can fit analytical models to describe how each of the coefficients vary as functions of ϑ. In the case of a_0, this might be a quadratic model $y = ax^2 + bx + c$ with the parameters a,b and c fit to the above experimental data. Now, given these models and if we assume a distribution for the angle ϑ for which reasonable choices might be that it is uniform or locally Gaussian, we can fully determine analytical distribution models for the coefficients by carrying out a simple transformation. Thus if, for example, $\vartheta \sim U(0, 1)$ then $y \sim$ Beta$(0.5, 1)$ and so on. Such descriptions of the probability distributions of the LCV coefficients could then be built into the chosen optimisation algorithm and used as sampling distributions, in order more efficiently to draw possible solutions from regions of high probability and spending little computational effort and time exploring regions of the vast, high-dimensional solution space that are unlikely to be relevant.

Finally, we point out that the form of the coefficients is to a large extent independent of the actual object and indeed the results presented here generalise[1] to any type of object (symmetric, asymmetric, convex or concave) under similar imaging conditions that is allowed to rotate about the chosen vertical axis between the basis views. It is possible to carry out similar experiments to characterise the effects of other 3-D rigid transformations on the LCV parameters. Although not examined here, under perspective projection the y-coordinates of the images of the landmark points will vary as the object is rotated as described above owing to the changing depth of points on the object. Hence, we expect the coefficients b_0, b_2 and b_4 to vary slightly as a function of ϑ. b_0 will have a similar quadratic form to that of a_0 and b_2, b_4 will linearly decrease and increase respectively.

Our treatment of the LCV coefficients (and their associated prior distributions) relies on their identified properties resulting from the isolation of individual transformations. These transformations span a high dimensional non-linear space (manifold) and isolating them in the way we did, amounts to only considering a single slice of this manifold at a time. Perhaps a more robust approach would be to use a low-dimensional embedding method that will allow us to learn the local properties of this manifold. Widely used examples are the Kernel PCA introduced by [Scholkopf et al. (1998)], which utilises an SVM to construct a non-linear mapping from the input space to a high-dimensional linear space. It has been used by [Gong et al. (2002)] to model the dynamic, non-linear changes in appearance (shape and texture) of an image accross a large pose angle variation. The Isomap by [Tenenbaum et al. (2000)] is

[1] Provided the objects are reasonably compact and are not seen from viewpoints improbably close to them.

another method designed to discover any non-linear degrees of freedom in high-dimensional data by using the geodesic distance induced by a neighbourhood graph to incorporate manifold structure in the resulting low-dimensional embedding. One example where it has been used successfully for classification is the work by [Yang (2002)]. Finally, Local Linear Embedding (LLE) [Roweis and Saul (2000)] is another option which attempts to discover non-linear structure in high-dimensional data by exploiting local symmetry of linear reconstructions and has been exploited to learn the appearance variation across face images [Mekuz et al. (2005)] and expression for face recognition [Liang et al. (2005)].

5.2 Bayesian model

In this section we extend the basic LCV equations (3.14) and (5.4) by incorporating prior information on the coefficients (a_i, b_j) and building a Bayesian model. We start with the Bayesian paradigm $P(x|d) \propto P(d|x)P(x)$ extended to n-dimensions:

$$P(\{x_1, x_2, \ldots, x_n\}|d) \propto P(d|\{x_1, x_2, \ldots, x_n\})P(\{x_1, x_2, ..., x_n\}) \tag{5.8}$$

expressed abstractly with x_i with $i \in \{1, \ldots, n\}$ as the unknown variables and d as the observed data vector. Now, if we assume that the x_i are statistically independent (5.8) becomes:

$$P(\{x_1, x_2, \ldots, x_n\}|d) \propto P(d|\{x_1, x_2, \ldots, x_n\})P(x_1)P(x_2)...P(x_n). \tag{5.9}$$

To apply this approach to the LCV method used as in equation (5.4) for the synthesis of an image I_S that we hypothesize should approximately represent or explain the target image I_T, we treat I_T as the observed data, the LCV coefficients (a_i, b_j) as the unknown parameters, the basis views, for which in this work there are just two: ${I_m}'$ and ${I_m}''$ as known a priori, and finally $\epsilon(x, y)$ as a vector of i.i.d. random noise [2] and $w' = \frac{d''^2}{d'^2+d''^2}$, $w'' = \frac{d'^2}{d'^2+d''^2}$ are the synthesis weights with $d''^2 = a_3^2 + a_4^2 + b_3^2 + b_4^2 + a_0^2 + b_0^2$ and $d'^2 = a_1^2 + a_2^2 + b_1^2 + b_2^2 + a_0^2 + b_0^2$. The posterior probability of the LCV coefficients given the target image I_T) thus becomes according to (5.9):

$$P((a_i, b_j)|I_T, {I_m}', {I_m}'') \propto P(I_T|(a_i, b_j); {I_m}', {I_m}'')P(a_i, b_j), \tag{5.10}$$

where $P(I_T|(a_i, b_j); {I_m}', {I_m}'')$ is the likelihood, that is the probability of observing the target image I_T given the coefficients (a_i, b_j) and also the basis view images ${I_m}'$ and ${I_m}''$. $P(a_i, b_j)$ is the prior probability of the LCV coefficients.

Since we are dealing with a high, n=10-dimensional space and although the posterior (5.10) is not normalised it will most likely numerically be very small when we are far away from the mode(s) in the tails of the distribution. This can cause approximation problems where the exponential is very close to zero because of the limited numerical precision of computers. It is therefore preferable to use

[2] As discussed previously, when the landmark points in the target image I_T are not correctly located, this last assumption cannot be completely correct. There will also be errors in the image synthesis caused by inaccuracies in the manual selection of landmarks and assignments of correspondences in the basis views during the off-line, model building stage that, although likely to be smaller than those just mentioned, nevertheless also mean this last assumption will not be completely correct.

the negative logarithm of the probability which alleviates this problem and, since the logarithm is a monotonic function, still maintains the global optimum at the same position. Hence, instead of (5.10) we use:

$$-\log[P((a_i,b_j)|I_T,I_m{}',I_m{}'')] = -\log[P(I_T|(a_i,b_j);I_m{}',I_m{}'')]-\log[P(a_i,b_j)+\texttt{''constants''}. \tag{5.11}$$

where the ''constants'' are independent of the LCV coefficients (a_i, b_j) and unimportant in finding the optimal values of the these coefficients. That suffices for our purposes but we note these terms would become important if we were also to optimise with respect to the variance and covariance parameters.

5.2.1 Likelihood

The likelihood in (5.11) is specified by the assumed probability density function (p.d.f.) of the fluctuations in the measurements about their predicted values and, strictly speaking the likelihood function should be based on the statistical properties of the noise. However, we may use the general assumption that the deviations ϵ of the synthesised image I_S from the target image I_T, are drawn from a multivariate iid normal distribution of covariance σ_ϵ^2. The log-likelihood is thus:

$$-\log[P(I_T|(a_i,b_j);I_m{}',I_m{}'')] = \frac{1}{2\sigma_\epsilon^2}\sum_{x,y}[I_T(x,y)-I_S(x,y)]^2, \tag{5.12}$$

which is quadratic in the residuals and the summation is carried out over all image pixels. The other term in (5.11) comes from the prior p.d.f..

We should note here that the independence assumption on the LCV coefficients is used to derive a tractable formulation for the posterior distribution and is not strictly accurate since we are dealing with an overdetermined linear system with more coefficients than degrees of freedom. In addition there is the implied independence on the pixel values which might not hold for highly correlated foreground regions. One way to achieve a form of pixel independence would be to filter the image similar to the work by [Sullivan et al. (1999)] so that the filter responses will be independent.

5.2.2 Prior

Recall the Bayesian interpretations discussed in section 1.5.1. Here, we use the latter, "subjective" interpretation where prior information comes from the analysis of the LCV parameters carried out previously. We can therefore use a Gaussian prior for the coefficients a_i and b_j centred at the positions already identified in section 5.1.4. Under the assumption of statistical independence between the coefficients, with each having its own mean and variance we obtain:

$$P(\{a_i,b_j\})$$

$$=\prod_{i=0}^{4}P(a_i)\prod_{i=0}^{4}P(b_i)$$

$$= \prod_{i=0}^{4} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-0.5\left(\frac{a_i - \bar{m}_a}{\sigma_i}\right)^2\right] \prod_{j=0}^{4} \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left[-0.5\left(\frac{b_j - \bar{m}_b}{\sigma_j}\right)^2\right]$$

$$= \frac{1}{(2\pi)^{5/2}\prod \sigma_i} \exp\left[-0.5\sum_{i=0}^{4}\left(\frac{a_i - \bar{m}_a}{\sigma_i}\right)^2\right] \frac{1}{(2\pi)^{5/2}\prod \sigma_j} \exp\left[-0.5\sum_{j=0}^{4}\left(\frac{b_j - \bar{m}_b}{\sigma_j}\right)^2\right]$$

$$= \frac{1}{(2\pi)^5 \prod \sigma_i \sigma_j} \exp\left[-0.5\sum_{i,j=0}^{4}\left[\left(\frac{a_i - \bar{m}_a}{\sigma_i}\right)^2 + \left(\frac{b_j - \bar{m}_b}{\sigma_j}\right)^2\right]\right]. \quad (5.13)$$

If we again ignore terms independent of the LCV coefficients which do not affect the optimal solution for these parameters, the negative logarithm of (5.13) may thus be written as:

$$-\log(P(\{a_i, b_j\})) = \sum_{i,j=0}^{4}\left[\frac{(a_i - \bar{m}_{a_i})^2}{\sigma_i^2} + \frac{(b_j - \bar{m}_{b_j})^2}{\sigma_j^2}\right], \quad (5.14)$$

where $\bar{m}_{a_i}$,$\bar{m}_{b_j}$ are the mean coefficient vectors and σ_i,σ_j the r.m.s. deviations of the prior probability for coefficients a_i and b_j respectively.

5.2.3 Posterior

The negative log of the posterior probability from (5.11),(5.12) and (5.14) becomes:

$$-\log[P((a_i, b_j)|I_T, I_m{}', I_m{}'')] = \frac{\sum_{x,y}[I_T(x,y) - I_S(x,y)]^2}{\sigma_\epsilon^2} + \sum_{i,j=0}^{4}\left[\frac{(a_i - \bar{m}_{a_i})^2}{\sigma_i^2} + \frac{(b_j - \bar{m}_{b_j})^2}{\sigma_j^2}\right]. \quad (5.15)$$

We usually require a single synthesised image obtained from a well-defined set of optimal LCV coefficients (a_i, b_j) to be presented as the result. A typical choice for that single image is the one which maximises the a-posteriori probability (MAP) or equivalently which minimises the negative log-posterior (5.15) with respect to the parameters a_i and b_j:

$$\min_{a_i, b_j}(-\log[P((a_i, b_j)|I_T, I_m{}', I_m{}'')]). \quad (5.16)$$

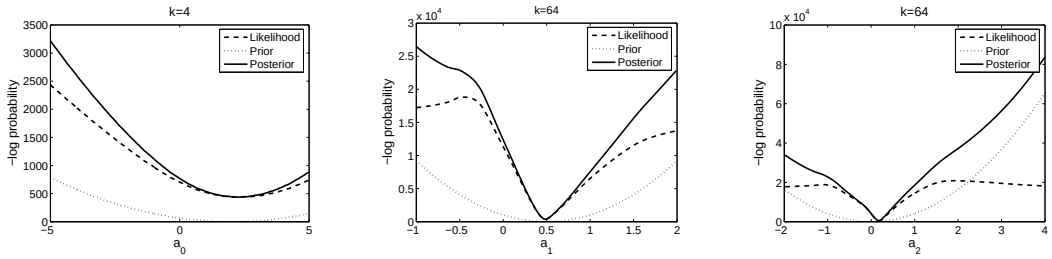
The above can be minimised using standard optimisation techniques.

As we can see from (5.15) the prior is used to bias the MAP solution towards the means $\bar{m}_a$ and $\bar{m}_b$ away from the maximum likelihood (ML) solution which is where $\sum_{x,y}[I_T(x,y) - I_S(x,y)]^2$ is at a minimum (i.e. there is little difference between I_T and I_S). How much the prior affects the solution in relation to that which would be obtained from the likelihood alone may be characterised by the quantity:

$$k = \frac{\sigma_\epsilon^2}{\sum_{i,j}(\sigma_i^2 + \sigma_j^2)}. \quad (5.17)$$

As the influence of the prior vanishes (i.e. σ_i, σ_j become very large and the Gaussian prior resembles a uniform distribution) the MAP solution approaches the ML solution. Careful selection of the variances $\sigma_\epsilon^2, \sigma_i^2, \sigma_j^2$ is therefore important.

The results of using such Gaussian priors to bias the posterior can be seen in Fig. 5.4. Here we



(a) (b) (c)

Figure 5.4: Negative log-posterior plots for 3 of the coefficients.

show three one-dimensional plots of the negative log probability of the likelihood, prior and posterior for the coefficients a_0, a_1 and a_2. These graphs were generated by isolating and varying each of the coefficients in turn while having conditioned the remaining coefficients to the optimal values identified previously. The image I_S was synthesised and compared to the target image I_T with the log probabilities recorded at every iteration. We used image examples from the COIL-20 database [Nene et al. (1996)]. The means required in each of the three priors were also selected at the identified optimal values for the coefficients a_0, a_1 and a_2 and the standard deviations were chosen as $\sigma_{a_0} = 0.5$ and $\sigma_{a_1} = \sigma_{a_2} = 0.125$ respectively. The standard deviation of the noise in the likelihood was set at $\sigma_\epsilon = 1$. We examine only these three coefficients here since the curves are quite similar for the remaining seven.

What we should note in particular from these examples are the effects of the prior on the likelihood, especially near the tails of the p.d.f. (where we have larger error residuals). The prior widens the basin of attraction of the likelihood curve resulting in an almost convex posterior that is much easier to minimise even if we initialise our optimisation algorithm far away from the optimal solution. On the other hand, where we have the maximum probability near the global optimum we wish the prior to have as little impact as possible in order for the detailed information to come entirely from the likelihood. This is so that we can allow for some small deviations from the most likely values for the coefficients as encoded in the prior means since every synthesis and recognition experiment will differ slightly, owing to noise, perspective camera effects and so on[3].

The extent to which the priors will affect the posterior distribution can be determined by choosing appropriate magnitudes for the ratios $k_i = \sigma_\epsilon^2/\sigma_i^2$ and $k_j = \sigma_\epsilon^2/\sigma_j^2$. Thus, for example for coefficient a_0 for which the likelihood is already convex we can use a fairly wide Gaussian prior without need to take much care as to where it is centred. In distinction, for the coefficients a_1 and a_2 the basins of attraction in the likelihood are quite narrow and much stronger priors are required. We note again how a good choice for these ratios can ensure that exact position of the global optimum at the bottom of the overall basin of attraction is determined by the likelihood alone. For example, in Fig. 5.4(b) the prior mean is set to $\bar{m}_{a_1} = 0.5$ but the posterior minimum is at $a_1 \simeq 0.48$ because this is also the location of the minimum in the likelihood term. This is the exact location we wish to preserve when we calculate the posterior distribution.

[3]We have seen in a number of experimental cases where we allowed such deviations that the synthesis similarity between I_T and I_S was much higher (and thus much lower error) than when we used a much stronger prior to bias the solution closer to the prior mean values.

(a) Scene view (b) Basis image 1 (c) Basis image 2

Figure 5.5: Synthetic data used for the testing of the LCV object recognition approach

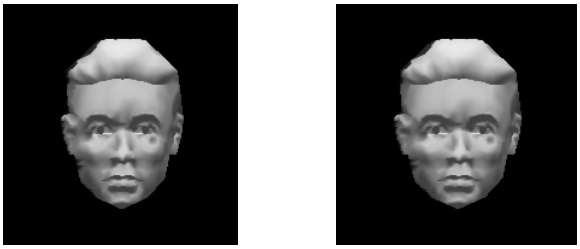
Essentially, we are proposing a flexible template matching system in which the template is allowed to deform in the LCV space but restricted by the Bayesian priors to regions where there is a high probability of obtaining meaningful solutions.

5.3 Experimental results

In a similar fashion to that adopted in the previous chapter, we present the results from a small number of tests designed to examine the validity of our 3-D object recognition method and particularly the Bayesian inference part. These tests will serve as a precursor to the more detailed experiments which follow in later chapters.

For these preliminary experiments we envisage the following object recognition problem which we will attempt to solve via the LCV approach. Consider the scene image of an artificial human head model (Fig. 5.5(a)) in a frontal-facing position in relation to the camera. We wish to identify this pose, here assigned an angle of 0^0, using a multi-view template model comprised of two given basis views. For the known basis views we chose two images (Fig. 5.5(b), (c)) that are $\pm 15^0$ apart from the frontal, scene or target view. We then built our LCV model by choosing 52 landmarks on prominent features of the object and carried out a constrained Delaunay triangulation that was kept consistent between the two basis views. With the help of a global optimisation algorithm (the details of which are not important at this point) we then examined three different examples: first, a search for the LCV coefficients by starting close to the optimum solution (i.e. a good initialisation); second, a similar search but starting from a remote location (i.e. a poor initialisation) and finally, the same case as used for the second, 'poor initialisation' experiment but with a Bayesian model available to regularise and localise the optimisation search. These tests were designed to give us some idea about the difficulty of the problem and form of the objective function and error surfaces, and also to illustrate, in practice, any beneficial effects of using the Bayesian approach.

We carried out 100 test runs for each example and every run was allowed to execute for 20000 evaluations of the relevant objective function. In total we thus performed 300 LCV object recognition tests for the recovery of the frontal view. The success of each run was determined from evaluation of two quantities. The first was the back-projection error $E_B = \sum_{i=1}^{N} d_i^2$. This is a purely geometric measure defined as the SSD between the landmark points in the scene or target image and the corresponding landmark points in the synthesised image as calculated from the LCV equations. The total number of



(a) (b)

Figure 5.6: Two synthesised examples at the chosen thresholds. (a) c.c=0.966 and (b) E_B=108

landmark points in any one image was $N = 52$. We refer to it below as the "back-projection" error. The second quantity was the cross-correlation between the target and synthesised images which combines information as to how well both the geometry of the landmark points and the pixel intensities were synthesized. The ground truth solution (allowing for a small amount of error inherent in the approximations in the LCV equations and in the way we computed the pixel intensities) is given by the LCV coefficient set: [$a_0 = -3.3405$, $a_1 = 0.5115$, $a_2 = 0.0005$, $a_3 = 0.5212$, $a_4 = 0.0005$, $b_0 = 0$, $b_1 = 0$, $b_2 = 0.5$, $b_3 = 0$, $b_4 = 0.5$] with a cross-correlation of 0.988106 and back-projection error of 13.5502.

Following the above experiments we chose the convergence thresholds for cross-correlation and E_B as $\tau_c = 0.966$ and $\tau_{E_B} = 108$ respectively which were chosen from qualitative inspection of the image synthesis results. Thus, if for example we visually compare two synthesised instances, one of which has a cross-correlation ≈ 0.966 (Fig. 5.6(a)) and a second with $E_B \approx 108$ (Fig. 5.6(b)), to the target image (Fig. 5.5(a)) we can see that the two models appear to provide a sufficiently close match. We thus regard a successfully synthesized image as one that has both a cross-correlation $\geq \tau_c$ and $E_B \leq \tau_{E_B}$. We deliberately avoided placing individual distance thresholds on the 10 coefficients since, in more practical scenarios, they are not statistically independent as we discovered for the parameters in the 2-D example in chapter 4. Furthermore, owing to the over-determined linear system (3.14) it might be possible to reach a good solution that is outside the boundary limits set on the variation of the LCV coefficient as determined in section 5.1.4. In fact, we have seen a particular occurrence of this in some of our experiments. Study of the diversity plot (Fig. 5.7) reveals that coefficients a_2 and a_4 are lying outside the identified boundaries with higher diversity than other coefficients. In spite of this, all the models produced by these values are still very good representations of the target image and thus admissible as correct solutions to the optimisation problem. Thus it is not the case that solutions outside the predefined limits are not useful. However, the opposite is always true in the sense that a solution found well inside these boundaries will produce a good visual representation and will be admissible under with respect to the thresholds τ_c and τ_{E_B}. Because of this choosing the Bayesian priors to exclude coefficient values outside these boundaries is possible.

The test runs with a good initialisation were started inside the boundaries with: [$\{-5 \ldots 5\}$, $\{0 \ldots 1\}$, $\{0 \ldots 0\}$, $\{0 \ldots 1\}$, $\{0 \ldots 0\}$, $\{0 \ldots 0\}$, $\{0 \ldots 0\}$, $\{0.5 \ldots 0.5\}$, $\{0 \ldots 0\}$, $\{0.5 \ldots 0.5\}$]. Note the very restricted ranges for the coefficients that remain constant during the rotation of the viewpoint (or object) about the vertical axis. For the examples that were started from a poor initialisation, we

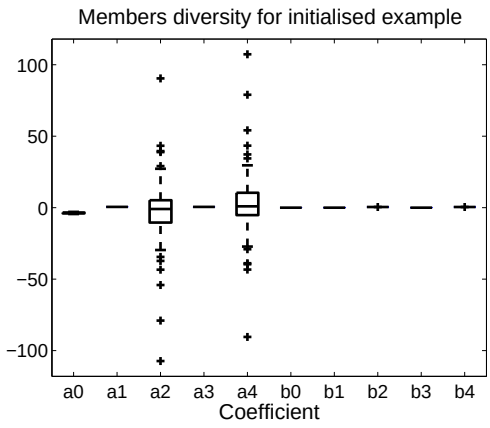


Figure 5.7: The diversity of the coefficients from the 100 tests with good-initialisation.

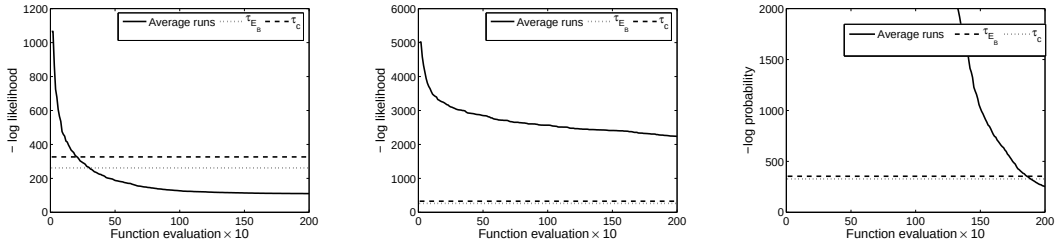
	No init	Init	Bayes
Total success %	0	100	96
E_B success %	0	100	96
c.c. success %	0	100	98

Table 5.1: Object recognition results for the 3 different cases.

defined the boundaries as: $[\{-5\dots5\}, \{-1\dots1\}, \{-1\dots1\}, \{-1\dots1\}, \{-1\dots1\}, \{-1\dots1\}, \{-1\dots1\}, \{-1\dots1\}, \{-1\dots1\}, \{-1\dots1\}]$. For the tests in which we used a Bayesian approach we kept the same boundaries as in the second set of experiments and used Gaussian priors with means and standard deviations: $\{m_{a_0}$=-2, m_{a_1}=m_{a_3}=m_{b_2}=m_{b_4}=0.5, m_{a_2}=m_{a_4}=m_{b_0}=m_{b_1}=m_{b_3}=0$\}$, $\{\sigma_{a_0}$=σ_{a_1}=σ_{a_3}=1, σ_{a_2}=σ_{a_4}=σ_{b_0}=σ_{b_1}=σ_{b_2}=σ_{b_3}=σ_{b_4}=0.01$\}$ for the 10 coefficients respectively.

The main results that show convergence of the optimisation for the three cases are assembled in Table 5.1. Here, we can not only examine each error measure separately but also see the combined results. It is obvious (column 3) that all the runs which were initialised close to the desired optimal or ground-truth solution not only converged successfully but also within a low number of function evaluations (see Fig. 5.8(a)). This most likely indicates a favourable region near and around the location of the global optimum location that lies within its basin of attraction. Provided that the optimisation algorithm manages to find its way into this favourable region we may then be able to achieve convergence to the globally optimal solution by using a simple, local optimisation approach.

On the other hand, the error surface far from the globally optimal solution is very difficult even



(a) Good initialisation (b) Poor initialisation (c) Bayesian

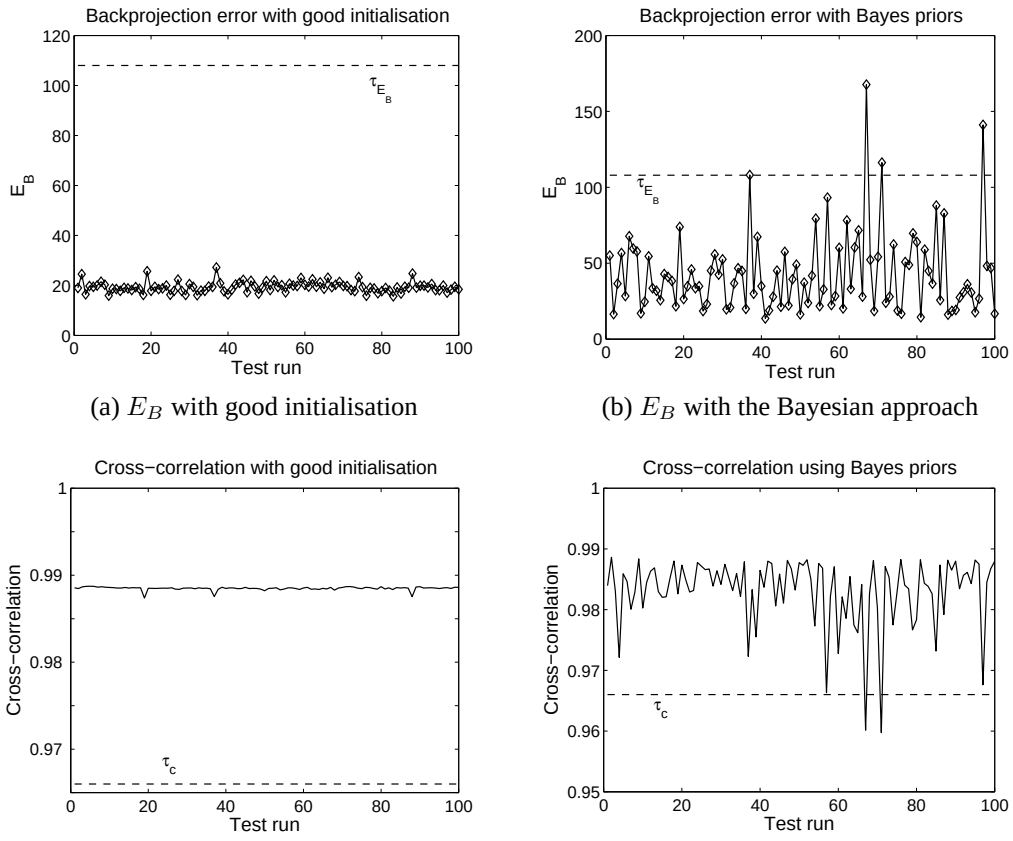
Figure 5.8: The average optimisation behaviour of the 3 examples.

for a powerful, 'global' optimisation algorithm successfully to traverse to the desired solution. We can see this in column 2 of Table 5.1. None of the 100 test runs in this column succeeded in finding the desired globally optimal solution and most did not get close to the optimum model configuration (see Fig. 5.8(b)). They either exhausted the allowed number of objective function evaluations or converged to spurious local optima. We may thus deduce that a way of successfully traversing these difficult and noisy regions of the parameter space is needed so that we can reach the correct solution efficiently, quickly and, most importantly, without getting stuck in local optima. This is exactly what the Bayesian approach aims to achieve by means of its regularisation and localisation effects. We can therefore use Gaussian priors to limit likely parameters values within the expected solution boundaries and simultaneously ensure they are not so strong that they overly bias the posterior. With such priors (See section 5.1.4.) we can achieve a similar effect to a good initialisation but with the diversity available for the optimisation algorithm to examine other promising areas of the solution space. In addition, the inherent smoothness of the Gaussian priors is incorporated into smoothing the posterior, especially in noisy areas as when the template is positioned over the image background, or in other words, in the tails of the distribution (see Fig. 5.4).

This behaviour of the priors is apparent from the runs of 100 trials in each of our experimental scenarios. The convergence results obtained from these runs are given in column 4 of Table 5.1. Here we see that the results of the Bayesian tests are almost as good as if we were to initialise close to the correct solution. In the tests of the Bayes approach, the algorithm was started at similar locations and with the same settings as in the poorly initialised cases just described but, because now the noisy background areas have been effectively smoothened out it managed effortlessly to converge to similarly (but not equally) low-error solutions as with the set of runs in the first case where a good initialisation was used (see Fig. 5.8(c) and comparison of the two error measures in Fig. 5.9). What should also be noted from Table 5.1 is that there is approximate agreement between the matching results as characterised by the two measures of cross correlation and back-projection error. This indicates that we appear not to have (or at least not to have discovered) any trivial solutions as were found in the 2-D affine example studied in the previous chapter. If we had such trivial solutions in which our model gives rise to an erroneous object representation, we would expect to see results with a high back-projection error but which, as in the 2-D case, had a low SSD error (or high cross-correlation). For such occurrences we would expect to see a big discrepancy between the 3rd and 4th rows of Table 5.1.

These preliminary tests have shown that the proposed object recognition paradigm using LCV is correct in principle and can be considered as an optimisation problem in the joint image space, similar to that for the 2-dimensional case examined previously. However, owing to the increased dimensionality we need to solve a more challenging optimisation problem and it has been demonstrated that a Bayesian approach which exploits our prior knowledge about the variation of the LCV coefficients is necessary when good bounds on the coefficient values to be used in the initialisation of the optimisation are not available.

A very desirable property of the LCV recognition method, that we identified from our initial tests



(a) E_B with good initialisation

(b) E_B with the Bayesian approach

(c) C. corr. with good initialisation

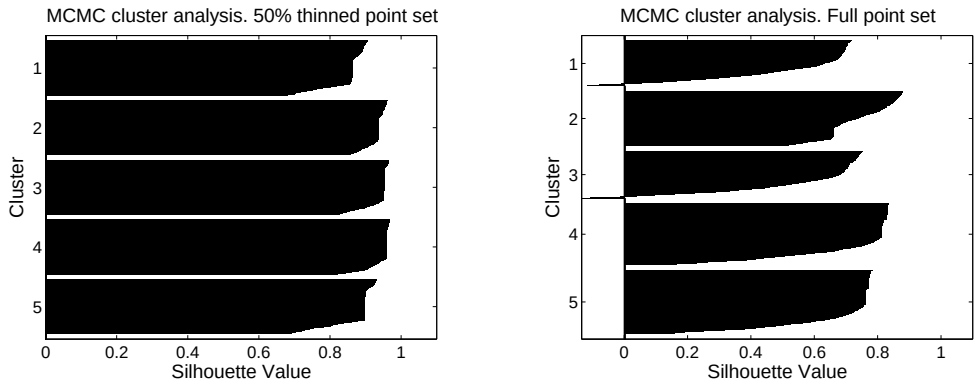
(d) C. corr. with the Bayesian approach

Figure 5.9: Comparison between the two measures for the good-initialisation and Bayesian tests.

but have not yet adequately proven, is that this approach does not seem to suffer from problems with trivial solutions. In order to make a more precise claim however it would be necessary to experiment much more extensively with additional transformations in 3-D that represent changes of viewpoint other than rotation about a vertical axis. We aim to do so in later chapters when we will carry out more detailed and structured experiments. With these preliminary results however we are confident of the validity and practicality of our method since it is obvious that a single, global minimum exists within a locally favourable area (that may be extended by the use of the Bayesian priors). We are thus simply faced with the (non-trivial) problem of efficiently and effectively reaching that minimum.

5.3.1 Markov-Chain Monte-Carlo

In the previous sections we have gone some way into providing general information about the overall shape and properties of the Bayesian posterior by specifying, up to constants and other irrelevant terms, a mathematical formula for the (log) posterior p.d.f. in (5.15) and by generating and visualising 2-dimensional slices of the objective function near the optimal solution. Helpful though the previous work has been, it is very desirable if we can obtain a better idea about characteristics of the posterior distribution more specifically relevant to the optimisation. We have therefore used Markov-Chain Monte-Carlo (MCMC) [Gelman et al. (1995)] sampling in order to generate a representative sample of the posterior p.d.f. from the regions of high probability and have carried out further numerical analysis on the distribution, since graphical analysis in 10 dimensions is not very feasible.



(a) (b)

Figure 5.10: The identified clusters before (b) and after (a) thinning-out the sample.

Markov-Chain Monte-Carlo (MCMC) is a general method for sampling from an unknown distribution that requires only that its density can be calculated at a sample point x, say. MCMC works by drawing values from a known distribution, the *transition distribution* and then gradually adjusting these draws to converge to the approximate posterior distribution (or *stationary distribution*). The samples are drawn sequentially with the draws forming a Markov Chain - that is - the distribution of the sampled draws depends only on the last value drawn. The method is driven by the transition distribution and some acceptance/rejection rule for the new samples In our implementation we have used the Metropolis-Hastings rule [Metropolis et al. (1953); Hastings (1970)] and a 10-dimensional Gaussian initial distribution in order to accept or reject new draws and begin the process of approximate the posterior distribution. In addition, in order to reduce any residual correlation between the drawn samples, it is commonplace to "thin-out" the samples by removing a subset (for example the first N samples) and keeping the remainder. This will also ensure that any bias from the initial transition distribution is greatly reduced.

We should emphasise here that MCMC is primarily intended to generate a sample from a distribution and is not an optimisation method. There is no guarantee that the MCMC can produce good point estimates. Although conventional importance sampling methods can be quite inefficient in high dimensional spaces MCMC is capable of reaching the areas of high probability, that is the main modes of a p.d.f., and drawing samples near or at such modes. Given the characteristics of our posterior distribution seen so far, it was decided to explore the MCMC both as a minimisation tool and as a mechanism for characterising the posterior p.d.f..

We chose the same object recognition experiment used in the previous section and generated a set of 10000 samples of the posterior (5.15) from areas of high probability using 5 Markov chains (2000 samples per chain) and with the following settings: standard deviation of the initial Gaussian distribution $\sigma = 10^{-5}$, initial acceptance probability $p = 0.95$ (that is when we start the algorithm, the initial Metropolis-Hastings criterion must evaluate to a probability of ≥ 0.95 for a sample to be accepted), acceptance ratio $r = 0.15$ (the percentage of samples that should be accepted in every N=10 samples drawn. The value of p is thus adjusted accordingly). As a starting point for the Markov chains we

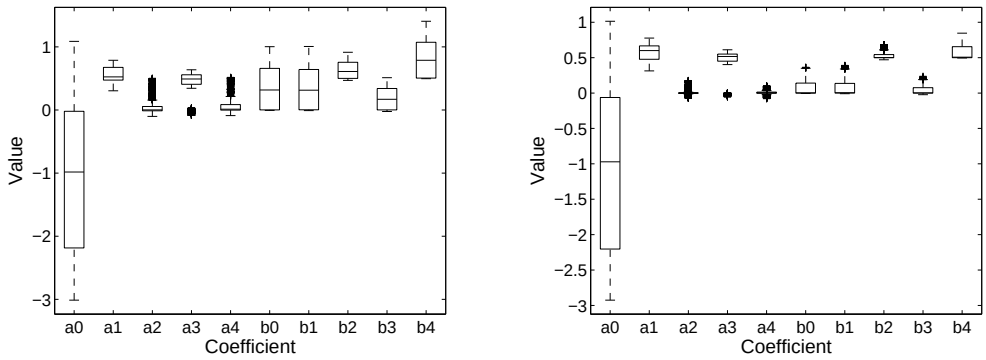
cluster 1:	-2.1893	0.7089	-0.0140	0.4685	-0.0005
(c.c. = 0.5792)	0.0779	0.1054	0.5297	0.0205	0.6061
cluster 2:	0.9493	0.5881	0.0544	-0.0306	0.0265
(c.c. = 0.1753)	0.1095	0.0994	0.5619	0.0621	0.5716
cluster 3:	-0.0765	0.4810	-0.0132	0.5455	0.0068
(c.c. = 0.7088)	0.0581	0.0458	0.5268	0.0431	0.5732
cluster 4:	-0.9653	0.3956	-0.0057	0.5776	0.0165
(c.c. = 0.7687)	0.0560	0.0524	0.4961	0.0219	0.5568
cluster 5:	-2.8922	0.6551	0.0125	0.5066	-0.0235
(c.c. = 0.6329)	0.0824	0.0684	0.5163	0.0478	0.5753

Table 5.2: The centres of the five identified clusters with their associated c.corr. values.

used similar bounds as examples from the previous section that were well-initialised. For analysis of the posterior we discarded the first half of the drawn samples (i.e. 1000 samples from each chain) while for the function minimisation we considered all the samples since the more samples available the better chance of one of them being near or at the global optimum. In fact, the MCMC method recovered a point very close to the global optimum with a cross-correlation of 0.97495 (the ground truth has cross-correlation of 0.9881 and the best solution recovered previously in the well-initialised tests was 0.9887).

For the analysis of the posterior based on the recovered, "thinned-out" sample, the first step is to determine any other major modes of the p.d.f. near and around the global optimum. That can tell us a lot about the shape of the p.d.f., especially where other locally optimal solutions may be situated. For that purpose, we used various runs of a k-means clustering algorithm [Bishop (1995)], the best of which recovered five main clusters (Fig. 5.10(a)) each associated with one of the Markov chains. The centres of these clusters can be seen in Table 5.2. It is obvious from the close proximity of the clusters and the fact that they are all near the global optimum, that the function has a single, main mode (i.e. peak) though with some noise which gives rise to other smaller peaks nearby, and that there is no significant local optimum elsewhere in the nearby posterior space. The fact that the centre of cluster 2 is far away in the value of a_3 coefficient merely indicates that the Markov chain failed to get very close to the global optimum and not that another significant mode is present. The presence of another significant mode would also have been identified by the Bayesian tests we carried out earlier. Note also that there is a greater diversity in the a_0 coefficient than in the others (see Fig. 5.11). This is to be expected since a_0 represents translation of the model along the x-axis and has different units (or as physicists say, dimensionality) from the other coefficients.

If, on the other hand, we do not thin-out the samples but consider all the 10000 points, including even those from regions of low probability, we also recover 5 principal clusters but in this case the clusters are not well separated (especially those with negative values, 1,2 and 3,4 Fig. 5.10(b)) most probably indicating a single, wide mode. From looking at the cluster centres and at the graph in Fig. 5.10 we did not discover any significant local optima which we expect, usually to be identified as clusters with high value (0.8,. . .,1) but with very thin footprint. Based on these clustering results we may say that the p.d.f. near the global optimum (which is of most interest to us) is a unimodal function, devoid of any significant local optima and affrected by only a small amount of noise as is to be expected since we



(a) (b)

Figure 5.11: The diversity of the 10 coefficients before (a) and after (b) the thinning-out.

Dispersion measures:						
	range	3.9392	0.4639	0.2480	0.6645	0.1798
		0.3701	0.3994	0.2119	0.2591	0.3503
	std. dev.	1.3890	0.1171	0.0407	0.2258	0.0302
		0.1031	0.1024	0.0455	0.0579	0.0973
	min.	-2.9253	0.3131	-0.0704	-0.0532	-0.0800
		-0.0062	-0.0085	0.4697	-0.0229	0.4959
	max.	1.0139	0.7770	0.1776	0.6112	0.0998
		0.3638	0.3909	0.6816	0.2362	0.8462
Location measures:						
	mean	-1.0348	0.5657	0.0068	0.4135	0.0052
		0.0768	0.0743	0.5262	0.0391	0.5766
	median	-0.9717	0.6004	-0.0003	0.5158	0.0000
		0.0026	0.0039	0.5013	0.0026	0.5067
	mode	-0.9826	0.4262	0.0000	0.5800	-0.0009
		0.0020	0.0000	0.5003	0.0000	0.5014
Distributional measures:						
	skewness	0.0558	-0.2909	2.0017	-1.3876	0.4002
		1.0680	1.2762	1.7307	1.3652	0.9414
	kurtosis (-3)	-1.3851	-1.0746	5.1036	0.1009	1.4513
		-0.2280	0.5369	2.3347	0.7000	-0.4407

Table 5.3: The results from the numerical tests on the drawn sample.

are dealing with discrete data.

One additional graphical tool that may be used to aid our analysis is the boxplot which illustrates the diversity of the coefficients in the samples from the MCMC. We have included two such plots, one prior to the thinning-out with all the points included (Fig. 5.11(a)) and the other after the thinning-out with only half of the sampled points (Fig. 5.11(b)). It is obvious that in the latter the samples are much more tightly compact with fewer outliers than when the data is not thinned-out. This is also as expected and is an indication that the algorithm has converged to an optimum location. Furthermore, this reinforces the notion that the posterior p.d.f. is unimodal leading to a narrow,and perhaps somewhat kurtotic, basin of attraction in the optimisation. In the first boxplot the existence of a large number of outliers simply illustrates that the algorithm has spent its initial time “randomly walking” through the high-dimensional

space of the LCV coefficients until it reaches an area of high posterior probability. The fact that there is lower overall diversity in the second boxplot shows that the removal of the first half of the drawn samples is a good way of reducing the dependence on the starting distribution while also limiting the presence of samples from regions of low probability in the tail of the p.d.f.. Note once again, as in Fig. 5.7) the increased diversity in coefficients a_2 and a_4 that represent correct solutions outside the identified boundaries.

We proceed with the calculation of the moments from the thinned-out sample as they may give us additional, numerical information about the properties of the posterior distribution. These are compiled in Table 5.3. Our first observation is that the mean, mode and median are in close proximity to each other, further reinforcing the evidence that we are dealing with an approximately symmetric, unimodal distribution (near and inside the basin of attraction). This is to be expected in particular owing to the effects of the prior which itself is a symmetric and unimodal distribution. By further examination of the range, minimum and maximum values, combined with the sample diversity box plot (Fig. 5.11), we can see once more how the coefficients are tightly concentrated within the general limits identified by the 3-D experiment described in section 5.1.4. This indicates a region of the error surface around the global minimum which is narrow and thin until it peaks out (or rather bottoms out) into a few close-by points. This limited spread, is further affirmed by the identified low standard deviation values in all 10 dimensions except for the coefficient a_0.

The last two numerical measures are the skewness and the kurtosis. These provide information about the asymmetry of the p.d.f. and the shape around its peak. As we mentioned above, the small numerical differences between the mean, mode and median may indicate an almost symmetric distribution. However, the skewness values in Table 5.3 demonstrate some positive skewness in certain dimensions, while there is negative skewness in others. This is mostly due to the shape of the likelihood function (i.e. the observed data) since the prior is symmetric. An example of the shape and skewness of the likelihood near the global minimum for some coefficients can be seen in Fig. 5.4. Finally we have the kurtosis of the peak which result from interplay of both the shape of the likelihood and the strength of the prior. For example, some dimensions have an almost Gaussian-like kurtosis of zero where there is little bias from the prior. Other dimensions however are highly kurtotic (leptokurtic) where the prior has greater influence that the likelihood and produces a narrower looking basin of attraction.

Even though we cannot visualise the 10-dimensional posterior p.d.f. we can say that as a product of the likelihood and prior distributions the posterior to some extent inherits characteristics of their shapes. Thus, it is unimodal and in some dimensions moderately positively skewed due to the shape of the likelihood and, depending on the strength of the prior we may have different levels of dispersion of samples drawn from the posterior. A highly biasing prior will produce a long, narrow p.d.f. while a weak prior will generate a shorter, wider peak in the posterior.

5.4 Summary

In this chapter we have seen how the linear combination of views (LCV) method may be used in view-based object recognition. Our approach involves synthesising intensity images using LCV and compar-

ing them to the target scene image. In addition we incorporated prior probabilistic information on the synthesis parameters by extending the LCV equations into a Bayesian model. For the priors, we chose Gaussian distributions centred around the identified locations of where the optimal synthesis parameters were expected to be. These locations were identified by isolating a specific transformation (in this case rotation about a vertical axis in 3-D) and interpreting the parameters as a function of the transformation.

We experimented with synthetic data and the use of an optimisation algorithm to recover the optimal set of parameters that would match the synthesised and target images. These initial experiments carried out in order to test the principle of our method while evaluating the advantages of using a Bayesian approach have shown that our method works well in recovering a view that lies between the basis views. Furthermore, we have seen the positive regularisation and biasing effects of carefully chosen priors on the matching objective error function and consequently on the optimisation results themselves. Finally, we used a MCMC to draw a sample from the posterior distribution and carried out additional tests in order to recover more information about the shape of the distribution near the optimal MAP solution and to probe where other interesting solutions may lie. The use of MCMC as an optimisation approach was also briefly explored with, because of the form of the posterior, satisfactory results. Nevertheless to revaluate the approach additional, more robust experimentation is required with a variety of datasets and across a range of different poses and objects. These are presented in the following chapters.

Chapter 6

Optimisation strategy

We have already seen a number of traditional and, for computer vision applications, novel optimisation strategies in Chapter 2. Our intention now is to test these different strategies against a set of 2-dimensional, analytic functions and real-image, realistic template-matching datasets. The aim behind these tests is to determine the general properties of each of the optimisation algorithms (using the 2-D functions) and understand some details about their parameter settings. We can then use this information and apply the same algorithms in a template-matching problem and see how they compare in more realistic circumstances and using real image data. This will give us further insight into the workings and parameter tuning of each method and determine which of these optimisation approaches best suits our kind of computer vision problem and data.

6.1 2-D test functions

The functions we will present here are designed to test the general properties of optimisation algorithms and give us an overall understanding of each method's strengths and weaknesses and possible parameter choices before we move on to datasets and experimentation specific to template matching. These functions were inspired by the work of [DeJong (1975)] and have been extensively used by optimisation researchers ever since to test the performance of various algorithms. The original set, comprised of 5 functions known collectively as DeJong's functions, include:

- the *sphere model*, $f(x) = \sum_{i=1}^{N} x_i^2$, a smooth, unimodal, symmetric, convex function used to measure the general efficiency of an optimisation algorithm. Since this function is very well behaved (from an optimisation point of view) the majority of standard, unsophisticated algorithms is expected to converge and we can use the number of function evaluations it takes an algorithm to reach the minimum as a measure of the algorithm's efficiency.

- *Rosenbrock's* function, $f(x) = \sum_{i=1}^{N} [(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2]$, which has a single global minimum inside a long, parabolic-shaped flat valley. To find the valley is quite trivial, however convergence to the minimum can be difficult. Algorithms that are not able to discover good directions for optimisation under-perform on this problem by oscillating around the minimum.

- *step* function, $f(x) = \sum_{i=1}^{N} \text{round}(x_i)$, which effectively highlights the problem of flat surfaces. Such surfaces pose particular difficulties for optimisation algorithms since they do not provide any

information as to which direction to favour. Unless an algorithm is equipped to handle variable step sizes then it can get stuck in one of the flat regions. Instead of the original step function, we decided to experiment with an alternative, the *six-hump camel-back* function, $f(x,y) = (4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2$ which has a wide and approximately flat plateau and a number of local minima. In addition, it has two, equally important global minima.

The camel-back function is more difficult than the original step function, since the flat region in the former does not offer enough information for a fixed-step algorithm to steer away from any local minima. Therefore, whereas in the case of the original step function an unsophisticated algorithm might search the error surface for a long time and eventually, purely due to luck converge at the global minimum, in the case of the camel-back function the flat surface near and around the local minima do not provide the necessary external energy for the algorithm to jump out and drift away to other promising regions. In other words, a combination of a flat surface surrounding local minima is more difficult to optimise than a flat surface alone.

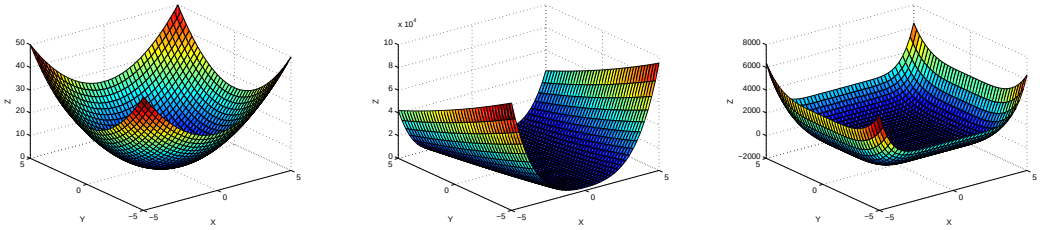
- *Quartic*, $f(x) = \sum_{i=1}^{N} x^4 + \text{Gauss}(0,1)$ is a unimodal function with the addition of random, Gaussian noise. This is used to test whether or not an optimisation algorithm can cope with noisy data. The problem with this function however is that the addition of a random part might shift the global minimum away from its known and expected location. This makes verification of the numerical convergence accuracy of an algorithm quite impossible. For this reason, we decided to use two alternative functions, *Rastrigin's* function $f(x) = 10n + \sum_{i=1}^{N} (x_i^2 - 10\cos(2\pi x_i))$ and the slightly more difficult *Griewank's* function $f(x) = \sum_{i=1}^{N} \frac{x_i^2}{4000} - \prod_{i=1}^{N} \cos(\frac{x_i}{\sqrt{i}}) + 1$. Both have a cosine modulation part to produce many local minima which although regularly distributed simulate the effects of noise (multiple modes) and most importantly do not change the position of the global minimum.

- The final function in the original set by De Jong was the *foxholes* function which contains many local minima. It is designed to test whether an algorithm can jump out of a local minimum or will get stuck in the first basin of attraction it encounters. We decided to use the aforementioned Rastrigin's and Griewank's functions for this test since they essentially serve the same purpose with the foxholes function.

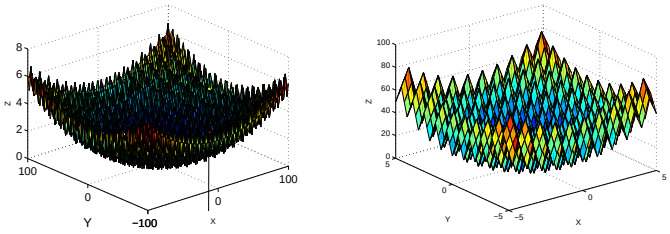
All the functions we will use for initial testing and evaluation of the optimisation algorithms are shown in Fig. 6.1.

6.2 Real-image template matching

In this section we propose more detailed experiments relevant to computer vision by examining deformable template matching since it is a generic scenario that might be applied to many different areas of interest in the field. The deformable template matching problem can be expressed as the task of searching for the parameters ξ of a transformation T that will bring the model template I_m into agreement with a target or scene image I_T. The model template may be represented in various different ways



(a) 1st De Jong's function. (b) Rosenbrock's saddle. (c) camel-back function



(d) Griewank's function. (e) Rastrigin's function.

Figure 6.1: The five 2-dimensional test functions.

such as using pixel intensities, feature points, edges, corners, linear segments and so on. The transformation T, for 2-dimensional problems, is usually an affine transformation with 6 parameters and may be mathematically defined in a similar way as in section 4.2 equation (4.6). In this case $g(.,.)$, our matching measure, is the sum of square differences dissimilarity metric where the sum is defined over all the features in the template, in this case pixels.

As a result we get the error surfaces for the 2-dimensional translation, anisotropic scaling and 1-dimensional rotation and shear as seen in Fig. 4.1. Of particular interest to us is the translation surface (Fig. 4.1(a)) because it contains the majority of problems confronting optimisation algorithms. This is due to the fact that, in general, a change in translation will move the model away from the object and on to the background region where unknown detail, background objects and clutter and thus more noisy peaks in the error surface exist. This is not so common with the other transformations. Thus the translation surface may vary depending on the type of template model I_m and scene image I_T we use. If for example we consider a template of the segmented object of interest and a scene image with the object present in front of a constant background (see Fig. 6.2(a)) then the translation space (assuming all other transformation parameters are optimally set) is a simple convex surface (Fig. 6.2(d)). It lacks any significant noisy areas (and thus local minima) and the global minimum may be easily found with even the most elementary of optimisation algorithms without the need for good initialisation. Though we note the changes in the error surface as detailed features begin to match, this is considered to be a relatively easy scenario of a computer vision optimisation problem and mostly encountered in controlled environments (e.g. assembly line visual inspection) and not so much with real images where considerably more noise and uncertainty may be present.

A second possibility is for the scene image I_T background to be substantially more complex (see Fig. 6.2(b)) with non-trivial structure and noise present. In this case however our template model I_m may be more elaborate also, composed of a full foreground and background model, or simply the foreground

object superimposed over the background. For this to work, we either have to know what the background is [Sim et al. (2002)], build a very simple model [Buxton and Zografos (2005)], or have a statistical model of what it is expected to be like [Srivastava et al. (2002, 2003)]. Therefore, for example in the case where a foreground/background model is available the matching error for when the template is over image background will certainly be higher than in the previous case (constant background) but will still produce a somewhat manageable translation error surface (Fig. 6.2(e)) since the background model will match over most of the background in the image. We consider this to be an example of a moderately hard optimisation task with most global algorithms and a number of local methods under good initialisation expected to converge to the correct minimum.

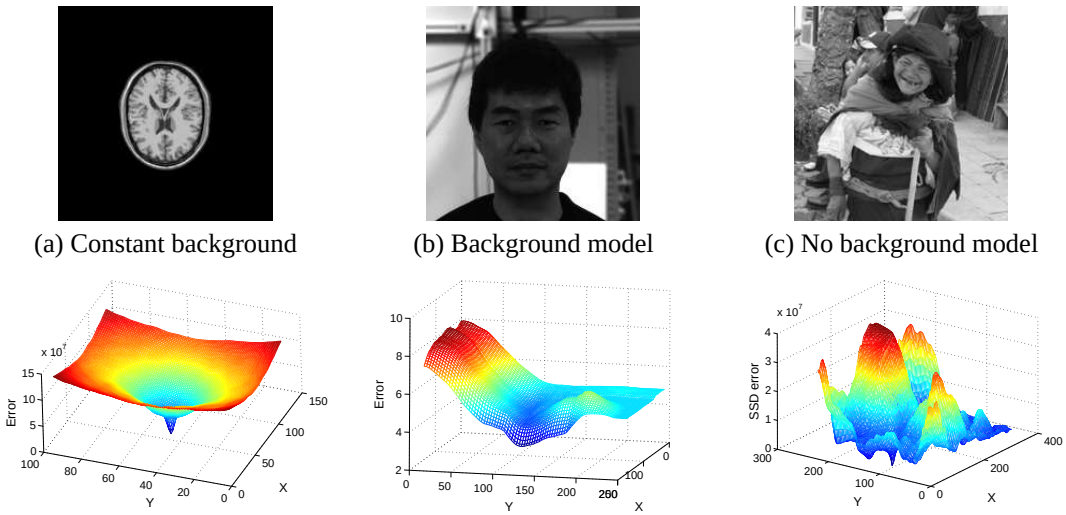
Finally, we have the hardest case where considerable structure and noise exist in the scene image background, but a model of the background is not available (see Fig. 6.2(c)). The optimisation difficulty in this scenario is apparent in the complexity of the 2-D translation error surface (Fig. 6.2(f)). We can see a "rugged" landscape with many local minima due to the noisy structure in the scene background and the absence of the regularisation effects of a background model. We note also that the global minimum is surrounded by a very narrow rim making the optimisation process even more problematic. In this scenario, all local optimisation methods not initiated in close proximity to the global minimum are expected to fail and most global methods will converge with great difficulty and after many iterations unless initialised appropriately and tuned specifically for this problem (i.e. boundaries, parameter settings, number of iterations and so on).

The importance of the inherent complexity of the translation error surface in the optimisation process has been demonstrated throughout many different test cases. If for example the translation parameters are kept fixed at optimal values, or if we initialise our search close to or inside the basin of attraction of the translational degrees of freedom, then all the global algorithms we have examined usually converge in all dimensions. In addition, unlike other parameters the translation space is usually[1] discrete and this introduces further problems to optimisation algorithms that cannot cope with a mixture of discrete and continuous parameters or that may require calculation of derivatives from a continuous function. Such problems may be solved to some extent by relying on interpolation techniques and numerical approximation of the derivatives.

Regarding the remaining dimensions of the search space we would like to draw attention to the irregularities of the 2-D scale space previously examined in section 4.4.1. Finally, the rotation and shear spaces can be easily minimised even though for the rotation space (see Fig. 4.1(c)) there may be a number of local minima at angular intervals of $\pm\pi/2$ depending on the rotational symmetry properties of the object. If these local minima are particularly pronounced they may cause local optimisation methods to get stuck.

It is quite possible (and often the case) that other important local minima exist elsewhere in the vast, multi-dimensional space formed when all the individual transformations are combined. Such regions are quite difficult to detect beforehand and may only become apparent when the optimisation algorithm is

[1] Unless sub-pixel accuracy is used.



(a) Constant background (b) Background model (c) No background model

(d) Constant background (e) Background model (f) No background model

Figure 6.2: Commonly encountered datasets and their corresponding translation error spaces.

running. The reason for this is that it is not feasible to visualise the full 6-dimensional space (for the 2-D affine transform). In spite of this we believe that isolating the surfaces the way we did helps us to get a general idea about the overall properties of a specific transformation and tune our algorithm appropriately in advance. Additional adjustments can only be carried out after test runs of the optimisation algorithm so that problems caused by these local minima are identified and dealt with.

We can therefore see that the typical computer vision task of deformable template matching is fraught with optimisation problems owing to the special characteristics of the objective function and the resulting error surface. It is thus important that the optimisation strategy we choose is suited for and can cope with these challenges.

6.3 Experiments: methods and results

In the previous section we have presented the different test cases against which we will evaluate the different optimisation strategies. In this section we will present the experimental method we propose to use for each dataset, the set-up of each algorithm, and the comparative results from which we aim to draw some conclusions about the fitness and efficiency of each strategy in relation to the typical computer vision problem.

6.3.1 Set 1: 2-D test functions

The single quantitative measurement we have used to distinguish between the different optimisation algorithms is the total number of function evaluations (FEs) required before convergence. This is because we consider NFEs to be a general and algorithm-independent way of judging the efficiency and obtaining an overall idea about the properties of each method. Convergence was defined as a recovered error minimum no greater than $\tau = 10^{-4}$ of the known global solution and found within the allocated optimisation budget (1000 NFEs for local methods and 10000 NFEs for global methods). We decided to increase the NFEs for the global methods since these in general require more time to converge and a direct compar-

Function	FE	x_m, y_m	$f(x_m, y_m)$	X_A, Y_A	$F(X_A, Y_A)$	Converged?
Sphere	26	-0.0043,-0.0034	3.098E-5	0,0	0	Y
Rosenbrock	70	1.0037,1.0066	8.089E-5	1,1	0	Y
Griewank's	1000	-3.14,-4.43	0.00739	0,0	0	N
Rastrigin's	516	(0.238,-0.241)E-3	2.288E-5	0,0	0	Y
Camel-back	30	0.0903,-0.7151	-1.03157	0.0898,-0.7126	-1.0316	Y

Table 6.1: The test results for the 5 functions using a reducing-step restarting simplex.

ison between local and global algorithms with the same number of FEs would be misleading. Instead we chose separately to compare each category of strategies. The threshold τ was kept fixed in all cases. Additionally, where possible we tried to use similar initialisation criteria for each method in order later to facilitate intra-category comparison with respect to this aspect of the problem.

We begin with the simplex algorithm which was always initialised from the same triangle with $A = (5, 5)$, $B = (5, 0)$, and $C = (-5, -5)$. We carried out 5 tests for each 2-D function (since there is the random restart part of the algorithm which produces different results at each run) and averaged the results. For each test function therefore we present a result that was most indicative of the average behaviour of the simplex algorithm. The results are shown in Table 6.1. $F.E.$ represents the number of function evaluations until convergence or termination, x_m, y_m are the coordinates of the found minimum point and $f(x_m, y_m)$ the function evaluation at that point. X_A, Y_A correspond to the known global minimum of the given function and $f(X_A, Y_A)$ is the global minimum value. We will use the same notation throughout these tests.

As we can see most functions have converged to the global minimum with a moderate number of iterations. We already mentioned that the simplex is not the most efficient amongst the direct search methods in discovering the best possible optimisation direction, something which is can be seen from the moderately high NFEs required to solve the sphere function. In the case of the Rosenbrock function the simplex again needs a significant number of iterations due to oscillations in the valley near the global minimum. However, these oscillations are not considerable and the simplex converges in the end without any problems. Furthermore, we see from the camel-back function that the simplex can cope with the uncertainty created by flat surfaces since it supports variable step sizes due to its expanding and contracting nature. It does however require some time to jump out of the local minima. Finally, when it comes to noisy surfaces the simplex is able to cope with some noise (as in the case of Rastrigin's function) because it can restart when stalled inside a local minimum. However, this requires a large number of restarts (jumps) which is reflected by the high NFEs required. As for Griewank's function the simplex cannot overcome the numerous and narrow local minima and cannot solve this function even if we considerably increase the available NFEs.

For the pattern search method, we run the same experiments using the following settings: starting point $X = (4, 5)$, polling of the mesh points at each iteration using the *positive basis 2N* [Audet and Jr. (2003)] method; that is, we computed the objective function at the mesh points to see if there is a point

Function	F.E.	x_m, y_m	$f(x_m, y_m)$	X_A, Y_A	$F(X_A, Y_A)$	Converged?
Sphere	81	0,0	0	0,0	0	Y
Rosenbrock	89	1,1	0	1,1	0	Y
Griewank's	1000	-3.14,-4.43	0.00739	0,0	0	N
Rastrigin's	81	0,0	0	0,0	0	Y
Camel-back	169	-0.0898,0.7128	-1.03163	-0.0898,0.7126	-1.0316	Y

Table 6.2: The test results for the 5 functions using a pattern search algorithm.

with function value lower than the current point. A mesh expansion factor of 2 (i.e. the algorithm multiplies the mesh by 2 after each successful poll) and a mesh contraction factor of 0.5 (i.e. the mesh is multiplied by 0.5 after an unsuccessful poll). The results for the same 5 test functions can be seen in Table 6.2. What we can observe from these results is that on average pattern search requires more FEs than the simplex indicating that it is not so efficient nor can it discover good directions (there are some considerable oscillations in the valley of the Rosenbrock function for example). However, it did find the exact location of the global minimum in most cases and managed to deal with noisy functions much more efficiently than the simplex, that is - it can jump out of local minima faster. However, even the pattern search had problems for a significantly noisy function such as Griewank's. For the flat, camel-back function the pattern search eventually converged but with considerably more iterations than the simplex indicating that the fixed mesh expansion and contraction factors were not adequate in cases where there is no information (improvement or deterioration) about the current function value.

We now come to the global methods with first the genetic algorithm. In this case the NFEs were increased to 10000 by setting the population and generation numbers to 100 each. The initial population was randomly generated from a $U(-5, 5)$ distribution. Although the algorithm we have used is quite generic in nature there is a large variety of different genetic methods available for testing [Holland (1992)] especially in the *selection* and *reproduction* stages. It was thus not practically possible to examine all the known selection and reproduction methods and their permutations. Nevertheless, amongst those we did test, on preliminary experiments, the *stochastic uniform* selection and the *scattered cross-over* reproduction functions provided the best results and therefore we used them throughout the rest of this work.

The stochastic uniform selection function arranges each potential parent in a line in which each parent occupies a length of the line proportional to the parent's scaled value. The algorithm samples this line at equal steps and allocates a parent depending on the section of the line it is sampling. For the scattered cross-over function, a random binary vector is created and where the vector is 1, genes from the first parent are selected, and where the vector is 0, genes are chosen from the second parent. The child is formed by combining the two genes.

The results of using the described genetic algorithm to optimise the 5 functions are presented in Table 6.3. The behaviour of the algorithm apparent from these results is that overall the genetic algorithm is quite inefficient and can get very close to but cannot go below the $\tau = 10^{-4}$ threshold at least within

Function	F.E.	x_m, y_m	$f(x_m, y_m)$	X_A, Y_A	$F(X_A, Y_A)$	Converged?
Sphere	4600	0.002,-0.003	1.623E-5	0,0	0	Y
Rosenbrock	10000	1.0535,1.1009	0.01131	1,1	0	N
Griewank's	8300	0.0098,0.0005	4.851E-5	0,0	0	Y*
Rastrigin's	10000	0.0017,0.0001	6.45E-4	0,0	0	N
Camel-back	10000	0.07993,-0.716	-1.03111	-0.0898,0.7126	-1.0316	N

Table 6.3: The test results for the 5 functions using a genetic algorithm.

Function	F.E.	x_m, y_m	$f(x_m, y_m)$	X_A, Y_A	$F(X_A, Y_A)$	Converged?
Sphere	1600	-0.0063,-0.005	6.449E-5	0,0	0	Y
Rosenbrock	2800	1.0074,1.0152	6.5936E-5	1,1	0	Y
Griewank's	2100	-0.006,-0.0173	9.248E-5	0,0	0	Y*
Rastrigin's	2300	(0.653,-0.27)E-3	9.9214E-5	0,0	0	Y
Camel-back	1900	-0.0893,0.7158	-1.0315	-0.0898,0.7126	-1.0316	Y

Table 6.4: The test results for the 5 functions using DE.

the limit of 10000 function evaluations. It will in fact converge in all cases if we increase the FEs limit since it was still making progress before the optimisation budget was exceeded. What should also be noted is the fact that GA can cope rather well with noise since it has found the minimum location in Griewank's function the majority of (but not all) times. It is therefore best to use the genetic algorithm for difficult problems with, if possible, inexpensive function cost where a high number of FEs would be justified.

We continue with differential evolution. For this we used similarly a population limit of $NP = 100$ and number of maximum iterations $itermax = 100$. The F and CR values [Storn and Price (1997)] were set to 0.8 and 0.5 respectively and we chose the $Best1Bin$ strategy because it converged most of the time. The soft boundaries of $[-5, 5]$ were also selected inside which we randomly initialised the first population. The test results are presented in Table 6.4. Here we see that DE performs much better across all functions and is more efficient than the GA. Even though DE is an evolutionary algorithm and needs to maintain a population of solutions (which equates to a high number of NFEs) it managed to recover the global minimum in all cases with a low NFEs especially in comparison to the maximum allowed NFEs. Furthermore, it succeeded in solving Griewank's function (albeit 80% of the times) which as we have already seen is a particularly difficult function which caused a lot of problems in all the optimisation algorithms discussed so far.

Finally we have SOMA, another example of a promising evolutionary method designed to solve difficult global problems. SOMA's parameters were selected as follows in order approximately to have a maximum of 10000 FEs: $step = 0.11$, $pathLength = 2$, $prt = 0.1$, $migrations = 50$ and $popsize = 10$. We also found that the best strategy in terms of average rate of convergence for this particular problem was the *SOMA all-to-one-randomly* strategy [Zelinka (2004)]. The initial population was initialised within the hard boundaries of $[-5, 5]$. Results of optimising the five 2-dimensional test

Function	F.E.	x_m, y_m	$f(x_m, y_m)$	X_A, Y_A	$F(X_A, Y_A)$	Converged?
Sphere	1302	-0.0085,-0.0049	9.54E-5	0,0	0	Y
Rosenbrock	10000	1.1159,1.2453	0.0134	1,1	0	N
Griewank's	10000	-3.14,-4.4384	0.0074	0,0	0	N
Rastrigin's	4570	3.29E-6,2.81E-4	1.577E-5	0,0	0	Y
Camel-back	2651	-0.0866,0.7136	-1.0316	-0.0898,0.7126	-1.0316	Y

Table 6.5: The test results for the 5 functions using SOMA.

functions using SOMA are given in Table 6.5. We can see that SOMA performs well on the sphere function indicating that it is quite efficient when used on simple test functions (as far as global methods are concerned). It can deal with a certain amount of noise (for example, it solves Rastrigin's function) but not with an overly complicated and very noisy function such as Griewank's. SOMA is also quite capable of coping with uncertain, flat regions by appropriately varying its step length when no more improvement is being made. It is not exceptionally good however in determining good search directions since it could not converge for Rosenbrock's function although it did come close. In short, we can conclude that in terms of general efficiency and optimisation performance SOMA lies between GA and DE, with GA being the least attractive of the global algorithms we examined.

As a result of these basic tests the best performing local optimisation method when comparing NFEs and average convergence was the reducing-step restarting simplex and, from the global methods, differential evolution. Before we can draw any broader conclusions however we need to perform more rigorous tests on real-image datasets.

6.3.2 Set 2: Real-image template matching

We shall further analyse the fitness of each of the examined optimisation algorithms by performing more detailed tests with the 3 real-image datasets previously discussed and described as: easy, moderate and hard, using a template matching objective function with 6 d.o.f.. In all the tests we aim to measure and investigate a greater range of the quantitative properties of each method so as to determine their convergence capabilities. We define convergence in this context as the ability to recover a model configuration (i.e. the 6 affine transform parameters) within some Euclidean distance threshold from the known optimum configurations. We could have also used the recommended minimum value to determine convergence, that is after the run to 'characterise or evaluate' how well the algorithm had converged, but in this case and especially when using a SSD dissimilarity metric it is quite possible to find an invalid model configuration with an error value that is lower than the expected global minimum, as we have mentioned already [Zografos and Buxton (2005a)].

The Euclidean distance is a much better way of judging how far away (and thus how much worse) we are from the optimal configuration since it does not suffer from these kind of problems. The only issue with using the Euclidean distance in a multi-dimensional parameter setting is that there must be a correspondence between changes in the parameters. For example, a change of one unit in translation should transform the model in an analogous way as one unit of change in rotation. This is not so

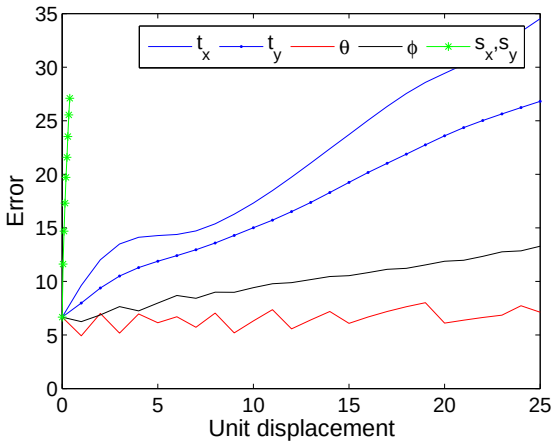


Figure 6.3: Comparison between parameter displacement and error response.

important for step-adjusting optimisation strategies that can automatically cater for this inequality but it is important for strategies that take random steps in different directions , and also for when we wish to analyse test results using the Euclidean distance of the transformations. How such transformations are measured or indeed defined is an open subject . One possibility would be to define model transform as the mean displacement of foreground image pixels such as the one used by [Studholme et al. (1996)]. We may argue that such a definition does not capture the disproportional changes in the calculated error that occur as the transformation parameters are varied. If for example we consider a change of 2 units in horizontal translation, this will not generate analogous changes in matching error as a 2 unit increase of horizontal scale. According to [Studholme et al. (1996)] the relationship between the translation and scale parameters is in the order of $4/F_x$ where F_x is the overlap between the scene and target images. However, if we use the error as the comparison basis (Fig. 6.3), we can observe that this relationship ratio is much higher.

A more practical alternative solution would be to normalise according to the effective range of each parameter. By effective range we signify the empirical boundaries for each parameter inside which the solution is expected to lie. Although this might work in practice it does not ensure that the individual transformation parameters are kept within these boundaries. In other words, it is possible for the 6-D Euclidean distance to be below an acceptable threshold but one or more of the transformation parameters not to be sufficiently close to its optimal value. For this reason, we decided to consider the individual 1-D distance for each of the parameters and impose proximity thresholds on each one separately. In this way, we do not have to be concerned with normalisation or that any of the parameters might be out of acceptable range.

The distance threshold boundaries were thus defined as follows, using some prior information about the expected effect on the error value: translation $t_x, t_y = 5$, scale $s_x, s_y = 0.1$, rotation $\theta = 10^0$, and shear $\phi = 5^0$. Any configuration within these limits from the known global minimum will be considered a valid solution and convergence will be deemed as successful. We used = the same values across all the 3 datasets.

Now that we have a definition of the convergence criterion we can define a number of different measures we may use to further analyse the characteristic behaviour of each optimisation strategy. Such measures are the *global minimum* of a *converged* test run; the *time to convergence*, that is how many iterations before the optimisation reached the convergence thresholds; the *convergence percentage*, that is the number of times the optimisation converged inside the set threshold; and the *diversity* in the recovered transformation parameters

Dataset 1 - MRI images

The first test data consists of an MRI scan of a human brain in front of a black background (Fig. 6.2(a)). A template of the object was generated from this image (i.e. similar lighting properties) and was subjected to a 2-D affine transform. We seek to recover the reverse of this transform that will bring the image and deformed template into registration. This transformation is: $(t_x, t_y) = 65, 68$; $(s_x, s_y) = 0.925, 1.078$; $\theta = -25$ and $\phi = -5.5826$. The dissimilarity SSD error between the optimal template and the scene is 0.0449 but because of additional interpolation and approximation errors, it is closer to 6.6689.

In all the tests that follow we try to maintain a fixed number of function evaluations: 2000 for local methods and 20000 for global methods, exhaustion of which would signify the end of a single test run. Every algorithm was allowed to perform 100 separate tests. None of the algorithms were initialised close to the ground truth solution but instead in order to maintain unbiased runs, they were initialised either far away and from the same starting point (for methods requiring a single initial value) or randomly within the parameter domains (for population based methods). In more detail, we used the following settings for each method:

- Simplex: initial restart step size S_0=[20, 20, 2, 2, 50, 20], cooling rate R=[0.95, 0.95, 0.9, 0.9, 0.9, 0.9], initial 7x6 simplex: fixed initialisation within the boundaries [1-50,1-50,0.5-1,0.5-1,1-20,1-20].

- Pattern search: initial random generated population in the range $(t_x, t_y) = [0 - 100]$, $(s_x, s_y) = [0.5 - 1.5]$, $\theta = [0 - 50]^0$ and $\phi = [0 - 10]^0$. Poll method = positive Basis 2N, polling order = consecutive, complete search = no. Initial mesh size = 30, rotate mesh = yes, scale mesh = yes, expansion factor = 2, contraction factor = 0.5.

- Genetic algorithm: 200 generations, 100 populations. Initial population function: random uniform in the range $(t_x, t_y) = [0 - 100]$, $(s_x, s_y) = [0 - 1]$, $\theta = [0 - 50]^0$ and $\phi = [0 - 10]^0$.

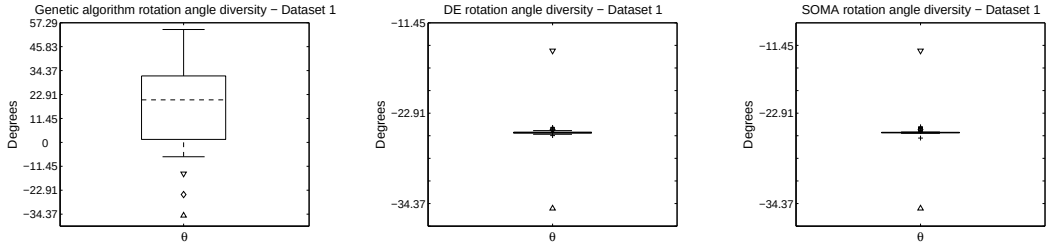
- Differential evolution: populations=100, maximum iterations = 200. F=0.8, CR=0.5, strategy=*Best1Bin*. Soft boundaries= $[1 - 100, 1 - 100, 0.5 - 2, 0.5 - 2, 0 - 100, 0 - 50]$.

- SOMA: step=0.5, pathlength=1.5, prt=0.1, migrations=100, popsize=50 $\cdots \approx 20000$NFEs. Hard boundaries= $[1 - 100, 1 - 100, 0.5 - 2, 0.5 - 2, -180 - 180, -50 - 50]$.

These settings will be kept fixed throughout all the datasets. After 100 experimental runs with each algorithm we obtained the following results for the MRI image dataset (see Table 6.6). In the second

	Dataset 1	Dataset 2	Dataset 3
DE	100% - 3915 FEs	96% - 889 FEs	61% - 11483 FEs
SOMA	100% - 2551 FEs	61% - 1416 FEs	97% - 4070 FEs
GA	0% - N/A	11% - 446 FEs	63% - 4603 FEs
Simplex	2% - 1060 FEs	2% - 476 FEs	1% - 1194 FEs
PSearch	12% - 476 FEs	3% - 0 FEs *	4% - 862 FEs

Table 6.6: Comparative results from the 3 datasets using all the algorithms.



(a) (b) (c)

Figure 6.4: The diversity of the rotation angle in the first dataset using GA (a), DE (b) and SOMA (c).

column we see the number of times the test runs converged inside the chosen distance threshold and the averaged time to convergence.

It is clear that both DE and SOMA have the best performance with all their test runs converging inside the threshold. DE uses only about 20% of the optimisation budget to achieve convergence on average but SOMA is the clear winner with approximately 1400 less FEs required for comparable results. Next we have the genetic algorithm which very suprizingly did not manage to converge in any of the 100 tests but instead converged inside one of the many pronounced local minima of the rotation parameter θ while having successfully identified the other parameters. We can see this from the high diversity in the recovered rotation angles (see Fig. 6.4(a)). This is due to particular symmetry properties of the human brain scan used as test object. The average recovered angle (horizontal dashed line) is much higher than the -25^0 ground truth (diamond shape) and well outside the $\pm 5^0$ threshold (up- and down-pointing arrowheads) fluctuating between $\approx -5^0$ and 55^0. DE and SOMA successfully manage to avoid this problem with a very low diversity in the final populations (see Fig. 6.4(b) and (c) respectively) well within the upper and lower angle thresholds of -30^0 and -20^0.

For the local methods, owing to the absence of good initialisation, we expect much lower convergence rates than the global methods. When the local methods are compared amongst themselves the pattern search can converge many more times and at around half the NFEs as the simplex requires. We also present a plot (see Fig. 6.5(a)) of the averaged, converged test runs for each of the above methods in order visually to compare the recovered minimum error and observe the representative optimisation behaviour of each algorithm. As expected, both local methods when they converge, do so much sooner (albeit on fewer occasions) than the global methods while the global methods find a good solution early and performance falls off gradually for the remaining allocated NFEs. We can see that in terms of the recovered minimum, DE and SOMA both have found a much lower solution than all the other methods which also is considerably lower than the practical ground truth (horizontal dashed line). This is

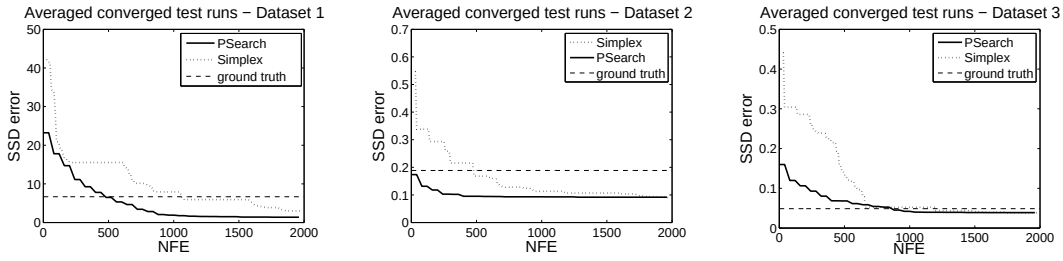


Figure 6.5: The average converged test runs for all the 3 datasets.

perfectly possible since the practical ground truth error includes approximation and interpolation effects which an optimisation method is able to counteract by appropriately adjusting the values of the system variables and thus reaching a lower error surface.

Dataset 2 - CMU PIE data

The second instalment of tests was carried out in a real image sample (see Fig. 6.2(b)) from the CMU PIE database with a complex background but which is given as a separate segmented image. This is a more difficult scenario than previously and we expect a lower convergence rate across all the methods. In this occasion, the practical ground truth is at $[82, 52, 1.0786, 1.1475, 10^0, -4.8991^0]$ with an SSD error of 0.1885 but as we mentioned above lower errors that correspond to good model configurations may be possible. The previously defined Euclidean threshold and algorithm parameter settings also hold in this case.

After 100 test runs for each optimisation algorithm we obtained the results in column 3 of Table 6.6. As expected we see an overall drop in the recognition results with DE being the dominant method with the best performance while at the same time displaying initial convergence behaviour reminiscent of a local method; that is, converging in under 900 NFEs. We can also see this in Fig. 6.5(b). The rest of the methods perform rather poorly with SOMA at 61% and GA at a much lower 11%. In the same graph we can also see that all three global methods exhibit a very similar optimisation pattern (at least in the test runs that converged successfully). Furthermore, all methods find a good minimum at ¡0.1, which is lower than the known solution. We also note that in the case of the pattern search algorithm the only 3 cases that succeeded in converging correctly were the ones that were randomly initialised inside the basin of attraction (see Fig. 6.5(b)).

Dataset 3 - Real image data without a background model

Finally we arrive at the hardest case; that of a real image with a complex background, but without any model of the latter (see Fig. 6.2(c) and (f)). Owing to the increased difficulty associated with this particular dataset it is expected that the overall optimisation performance will be further reduced. The optimal solution in this case is $[106, 59, 0.9048, 1.0444, 12.02^0, 0^0] = 0.0488$. If we use the same optimisation settings as previously we get the following results after 100 test runs (Table 6.6 column 5). SOMA performs very well with a 97% convergence ratio, with the GA coming second at 63% and DE not particularly efficient with this dataset at 61%. We also see that it takes DE many more iterations in

order to converge whereas SOMA and GA on average reach the global minimum around 2.5 times faster. Despite that all the global methods reached approximately the same minimum error. This is illustrated in the plot in Fig. 6.5(c).

In conclusion we may say that both DE and SOMA perform consistently well in all the 3 cases with an expected performance penalty associated with the increased difficulty of each dataset. Both these methods exhibit very low diversity of the parameters defining the optimal solution with them always inside the defined threshold and no outliers in the 6 coefficients across the 100 test runs, two properties that are very desirable for an optimisation algorithm. Another characteristic of their equivalent performance is the fact that they both reach approximately the same minimum at the end of their allocated FE budget. Where they differ however is in the time they require for initial convergence with SOMA being the clear winner since it manages to approximate the correct solution much earlier than DE (see Fig. 6.5). This makes SOMA ideal for the hybrid approach to be discussed later since we are able to switch to the local method much earlier in the optimisation process than with DE. As far as the GA is concerned, we have seen that when it converges successfully it can reach an equally good minimum error as obtained by SOMA and DE. Nevertheless, it has the tendency to get stuck in pronounced local minima for all but the simplest datasets which consequently reduces its effectiveness and thus it does not constitute a reliable algorithm for template matching-based object recognition. The two local methods, simplex and pattern search, can converge very fast and nearly to the same minimum whenever they reach its proximity. We can therefore use either one for the hybrid approach to be described next.

6.3.3 Hybrid approach

The hybrid approach is essentially the combination of a global, stochastic algorithm (in this case SOMA) designed to get us close to the basin of attraction as early as possible from a random, distant location on the error surface, and a local method (the simplex) whose purpose is rapidly to refine the good solution the global algorithm already recovered, much faster and more efficiently than the global method alone can. Ideally we wish to bring together the advantages of both the approaches in a manner that should neutralise their individual shortcomings. Specifically, those shortcomings are the slow and FE-intensive progress of the global method and the requirement for good initialisation and sensitivity to minima of the local approach. If we were to plot the average test runs of such an ideal hybrid algorithm we would expect to see an initial drop of the discovered minimum caused by the global method followed by a secondary drop due to the refinements of the local method instead of the gradual fall-off in latter part of the calculation traditionally associated with global, stochastic optimisation algorithms.

The only additional issue with using a hybrid method is how to determine when it is best to switch between methods. One possibility is to use a number of concurrent criteria to decide when we are close to the switch point. The first such criterion could be a proximity threshold such as the Euclidean distance previously used to determine convergence. When near that threshold, we many assume that the global optimiser has reached the global minimum and use the local method for further refinement. This threshold of course must be known before hand and thus may only be used when we are dealing with similar datasets of approximately the same convergence complexity or repeatedly running tests on the

same dataset for evaluation purposes (as in this case).

Another such criterion could be the observed relative gain $\Delta\epsilon/\epsilon$ of each successful iteration. When the gain is below some predetermined value we can assume that the global algorithm has almost stalled and switch to the local method with the expectation that it can burrow further into the error landscape.

A third criterion might be the relative change of each parameter $|\Delta p_i/p_i|$ at every iteration. When the change of the value in the parameters is insignificant at subsequent iterations then we may assume that the diversity of the population is very low and a change of optimisation approach (i.e. to the local method) might be necessary for further improvements to be made.

Alternatively, we may opt to use a fixed FE-related threshold based on the information we have about the optimisation behaviour of SOMA for that particular dataset. If for example we revisit Table 6.6 we can see that on average and across all 3 datasets SOMA requires between 1500-4000 FEs to reach the minimum error threshold. We can therefore use this prior knowledge and set SOMA to run at a fixed number of 4000 NFEs. Such a number will most. This again assumes some previous knowledge about the expected solution and is therefore limited in practical applicability.

As a result, we will use the following settings for the hybrid algorithm:

- SOMA: step=0.5, pathlength=1.5, prt=0.1, migrations=20, popsize=50 $\approx$ 4000NFEs. Hard boundaries= $[1-100, 1-100, 0.5-2, 0.5-2, -180-180, -50-50]$, method = All-to-one-randomly.

- Simplex: initial restart step size S_0=[20, 20, 2, 2, 50, 20], cooling rate R=[0.95, 0.95, 0.9, 0.9, 0.9, 0.9], initial 7x6 simplex that includes the vertex $V_{i,1}$ of the recovered system variables at the 4000^{th} function evaluation of SOMA and 6 random vertices $V_{i,2-7}$ generated at distance $d = [5, 5, 0.1, 10^0, 5^0]$ (note this is the Euclidean distance threshold from the previous tests) from the vertex $V_{i,1}$.

We carried out 100 test runs of the hybrid method for each of the 3 datasets (see Fig. 6.2) and we present the results in Table 6.7. The second row shows the convergence rate of the hybrid method. The percentage difference ($\pm\%$) in this row are in relation to the original SOMA results (row 2 of Table 6.6). The next two rows show the average SSD error of the 100 hybrid runs and the original 100 SOMA runs at 6000 FEs. The percentage differences of row three are in relation to the original SOMA results at the same NFEs. Finally, the last row shows the average SSD error of the original 100 SOMA runs at the maximum 20000 FEs, with a percentage difference in relation to the original SOMA error at 6000 FEs (row 4). We see that the convergence ratio is only around 15-30% lower than in the original tests but the error at 6000 FEs is between 20-65% lower than the error at the equivalent NFEs of the SOMA-only approach used previously. In fact, the error values are quite close to the original recovered minima using the full 20000 FEs. This can also be seen in Fig. 6.6. In these plots we can clearly see the secondary drop in the discovered minimum value due to the local method as we have mentioned previously and observe that the simplex algorithm always manages to refine the optimisation further (i.e. there is no stall at the switch point) indicating that on average we chose good switch points and that the local method can converge faster that the global method in the same number of iterations.

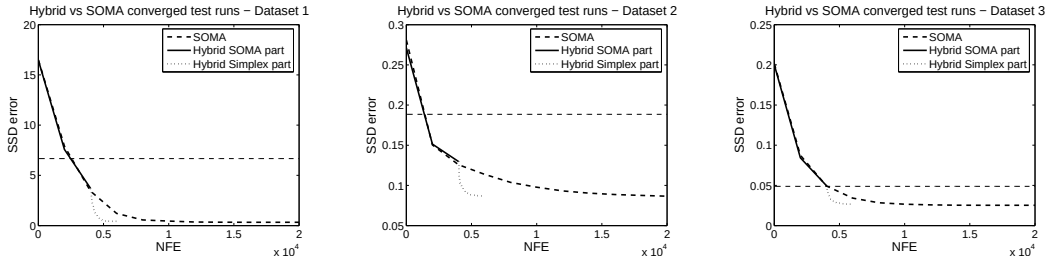


Figure 6.6: Plots comparing the hybrid approach and the SOMA method for the 3 datasets.

	Dataset 1	Dataset 2	Dataset 3
Convergence % ($\pm\%$)	86% (-14%)	41% (-33%)	81% (-16.5%)
Hybrid SSD @ 6000 FEs ($\pm\%$)	0.4275 (-65%)	0.0868 (-24%)	0.02661 (-22%)
SOMA SSD @ 6000 FEs	1.215	0.1138	0.03419
SOMA SSD @ 20000 FEs ($\pm\%$)	0.3265 (-73%)	0.08659 (-24%)	0.02523 (-26%)

Table 6.7: The results of the hybrid and SOMA tests at 6000 and 20000 FEs.

We can therefore say that by using a hybrid approach it is possible to obtain solutions that are very close to those obtained with a global algorithm alone but at a considerably reduced FE cost. In that sense a hybrid optimiser might be useful in situations where we are faced with a costly objective function but the good initialisation required for a local method is not available. With the application of the hybrid method we may in the early stages use a global algorithm to overcome the need for a good initialisation while avoiding the increased FE overhead due to its inefficiency in later stages of the computation. As we have already mentioned, switching between global and local methods is very important and so the effectiveness of the hybrid approach depends on the correct determination of this switching point.

6.4 Summary

In this chapter we have examined the task of deformable template matching cast as an optimisation problem. This is a particular challenge, ubiquitous to computer vision owing to the problem's generic nature and well-known difficulties. To address these difficulties, it was necessary to examine various optimisation methods (both local and global) that have not been adequately tested in this specific scenario in the past. In our work such traditional methods as the simplex, pattern search and genetic algorithm have been examined closely and compared to traditional global optimisation methods such as GAs and to methods apparently new to computer vision such as SOMA and differential evolution, the latter two having been originally applied to engineering problems.

We have tested the various approaches against a series of 2-dimensional, analytic functions designed to highlight the generic properties of each optimisation method (such as efficiency, discovery of good directions for the optimisation, sensitivity to noise etc), followed by three realistic datasets of progressive difficulty commonly encountered in computer vision. Their purpose was to determine how well each algorithm copes with typical template matching scenarios.

Our results show that the novel methods outperform the traditional global optimisation approaches

while being easier to set-up initially. The most promising method in terms of convergence, minimum error recovered and NFEs required was SOMA and therefore is the algorithm we will be using for our LCV experiments in the next chapter. Finally we argue that for this application a hybrid combination of a global and local method can produce equally good results in a fraction of the time required by a global method alone. We demonstrate this with a number of additional experiments.

Chapter 7

Experiments and evaluation

In this chapter we introduce a detailed evaluation of our LCV 3-D object recognition paradigm starting with the introduction of the various datasets used for testing, followed by the specifics of the experiments themselves, and concluding with a critical discussion of the test results. In addition, we examine an alternative, existing approach (Active Appearance models (AAM) by [Cootes et al. (2001)]) that aims to solve the same problem, and compare it with our method in order to determine just how well the LCV method fares against a tried-and-tested, well known technique. We end this chapter with the conclusions we drew from the results generated during the evaluation process.

7.1 Image datasets

In order to carry out our detailed evaluation experiments we have used three different datasets, consisting of synthetic and real-image examples. All three databases were generated via different methods and under various conditions, and are therefore quite different in size and content, but all of them include examples of objects imaged under varying pose, which is the principal focus of our work. The idea behind using a number of different datasets is to demonstrate the general validity of our results and the applicability of our method across a variety of cases. Of course, owing to the diverse levels of data complexity between the sets, we do not expect to recover the same quality of optimisation results, but as long as there is a graceful and predictable deterioration in the convergence outcome (i.e. see chapter 6, section 6.2), then we can assume that our models and algorithm are generally valid and robust. Because a model is tied to a particular dataset, to a certain extent, it does reflect some of the characteristics and complexities of that dataset, but in an obvious and manageable way.

$(-25^0, 0^0)$ $(0^0, 0^0)$ $(25^0, 0^0)$ $(0^0, -10^0)$ $(0^0, 10^0)$

Figure 7.1: Typical samples from the synthetic database at various rotation angles (hor.,vert.)

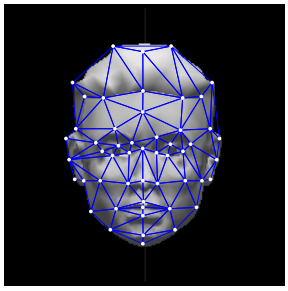


Figure 7.2: Synthetic database sample, showing the landmark points and Delaunay triangulation.

7.1.1 Database 1: Synthetic dataset

The synthetic dataset was generated using a 3-D head model by [Loizides et al. (2001)], which itself derived from [Parke and Waters (1996)]. The 3-D head model was projected onto a plane (using orthographic projection) and two dimensional synthetic face images were formed within a view range that maintained the visibility of all the landmark points in all the images. Namely, images generated by vertical axis rotation of the object between -20^0 to 20^0 from the frontal view (denoted here as 0^0), and at 5^0 intervals. Just as before we chose 52 landmarks from the subset of model vertices in order to minimise the approximation error (see Fig. 7.2). Additionally, we experimented with a few images outside the visible landmark range, at -25^0, 25^0 and also generated 4 images by rotation about the horizontal view axis at angles $\pm 5^0$ and $\pm 10^0$ from the frontal view, in order to test the extrapolation capabilities of the LCV model outside the range of the basis views and when some landmarks are occluded. In total, we used 15 pose samples, examples of which are shown in Fig. 7.1.

Furthermore, for all the eleven samples on the horizontal axis ($-25^0, \ldots, 25^0$), we generated 2 more distinctive expressions (happy and angry, see Fig. 7.3(a) and (b)) to test how well the LCV model can recover the optimal pose configuration in the presence of localised and limited deformations that were not (and cannot be) modelled by the LCV equations. In a more realistic scenario, such deformations might be the result of a change of expression. In addition, we introduced two different levels of random Gaussian additive noise in the pixel values in each of the above 11 samples (see Fig. 7.3(d)) to examine the robustness of the model and optimisation algorithm, when there is noise in the scene view but not in the basis views (i.e. it has not been modelled). Finally, we wanted to test against the effects of unmodelled limited occlusion, and thus randomly placed a circular object in front of the scene object (see Fig. 7.3(c)). We considered two possibilities; a foreground object with area equal to 20% of the head model and a foreground object at 40%. As such, this database as a whole contains 301 image samples. Details of the experiments performed on particular subsets of the synthetic database are given in later sections.

7.1.2 Database 2: COIL-20

The Columbia Object Image Library (COIL-20) [Nene et al. (1996)] is a database of gray-scale images of 20 objects. It was generated by placing the objects approximately in the centre of a motorised turntable and against a black background. The turntable was rotated through 360^0 about the vertical axis to vary

(a) Angry (b) Happy (c) 40% occlusion (d) $\sigma = 0.1$ noise

Figure 7.3: Synthetic samples with different expression, noise and occlusion levels.

Figure 7.4: Image samples from the COIL-20 database.

the objects' pose with respect to a fixed camera and under ambient (fluorescent) room lighting, in order to avoid strong shadows. Images of each objects were taken at 5^0 intervals, corresponding to 72 images per object, around the horizontal great circle of the view-sphere. The objects have a wide variety of complex geometric and reflectance characteristics and are shown in Fig. 7.4. In total, the database contains 1440 size-normalised and histogram-stretched images of the 20 objects.

Similarly to the synthetic dataset, we tested against the pose ranges between -20^0 to 20^0 from the frontal view at 0^0. Furthermore, we examined the two views at -25^0 and 25^0 outside the trained range of views, and where some landmarks were not visible owing to self-occlusions. In that case, the system had to cope as best as it could by extrapolating the required pose information. Landmarks where chosen along the main discontinuity boundaries of each object, and because the database contains more than one object the number of landmarks was different at each case. A typical example of an object with its landmarks visible, can be seen in Fig. 7.5.

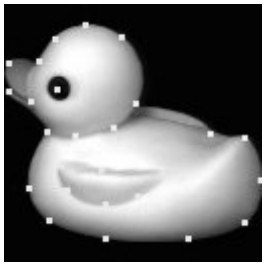


Figure 7.5: A typical sample from the COIL-20 database with chosen landmarks visible.

Figure 7.6: All the 10 individuals in the Yale face database B.

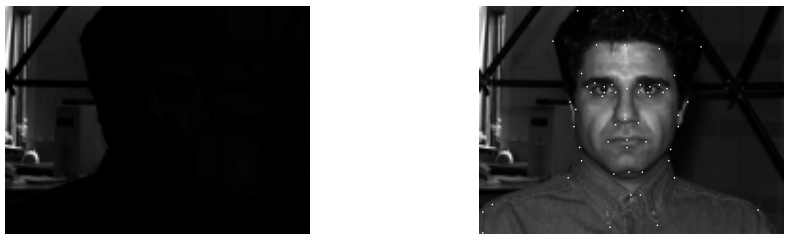
7.1.3 Database 3: Yale Face Database B

The Yale face database B [Georghiades et al. (2001)], contains 5760 single light source, grey-scale images of 10 individuals (Fig. 7.6), each seen under 576 viewing conditions (9 poses $\times$ 64 illumination conditions). As we can see in Fig. 7.8, the pose variations occupy a rather small portion of the view-sphere on the left of the frontal pose (number 0 at 0^0). More specifically, poses 1, 2, 3 and 5 are approximately 12^0 from the frontal pose and poses 6, 7, and 8 approximately 24^0. From those, poses 7 and 3 are taken in the same level as the frontal pose, while the rest are slightly above or below as arranged in Fig. 7.8.

For every individual in a particular pose, an image with an ambient (background) illumination was also captured. Note, this is not the same as a background only image (as in the case of the CMU PIE database) since the outline of the face is still visible (see Fig. 7.7(a)), but it may still help to regularise the search over background regions. In addition, the background is not strictly consistent between scene view and ambient illumination view, with people and objects appearing and changing position in the rear of the scene. Furthermore, the appearance of the background objects is somewhat influenced by the strong strobe lights used to illuminate the foreground object during the imaging process. This is one further problem with which our recognition system has to cope.

The images were captured using a purpose-built illumination rig, fitted with 64 computer controlled strobe lights. Images of an individual in a particular pose were acquired at a frame rate of 30 f/sec in order to minimise any unintentional discrepancy in pose and facial expression between the 64 images. The strobe lights were switched off for the capture of the images with the ambient illumination.

For our tests, we used all the available pose samples since the covered angle range is quite small and so the majority of the landmark points (see Fig. 7.7(b)) were visible in every view and all the images are from approximately the same aspect. Therefore, it was quite possible adequately to reconstruct all the images for every individual given an optimal choice of basis views. In other words, it was possible to reach all the views in the joint-image space. Furthermore, we chose to test a few examples of illumination variation for the frontal pose of a randomly chosen individual to see how well our system can cope with localised, non-affine changes in pixel intensity.



(a) (b)

Figure 7.7: (a) sample background in the Yale database and (b) sample landmark points.

Figure 7.8: All the different pose angles in the Yale face database B.

7.2 Training

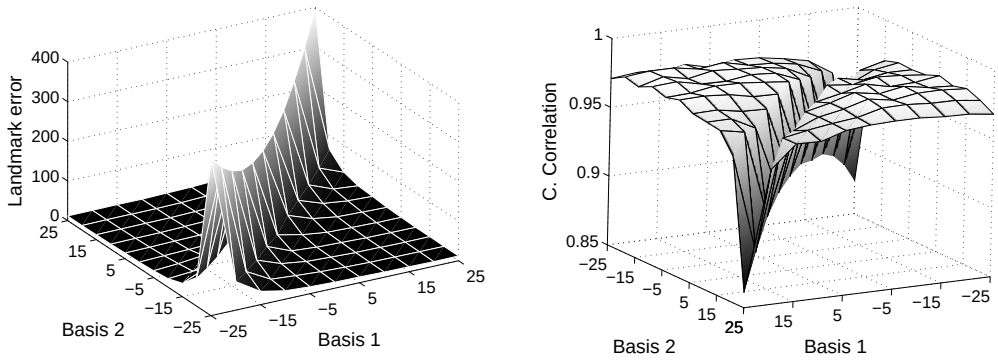
In this section we will discuss the general training method that we have employed in order to fine tune our models given a specific dataset. As we have mentioned earlier in chapter 5, our proposed recognition system includes an off-line modelling part, in which information about a 3-dimensional object is encoded into the system. We accomplish this by means of a small number of basis views, a set of corresponding landmark points consistently triangulated and previous knowledge about the synthesis coefficients, their range and distribution and probable configurations of the object built into the pdf component of a Bayesian inference mechanism.

From the above, the selection of appropriate basis views and the choice of prior distribution parameters are the only elements that change during training of a new model given an existing selection of stored images of an object. For choosing the basis views during the training of each model, we considered all the possible two-view combinations amongst the images in the training set (which as we shall explain later does not overlap with the test set) and calculated two separate RMS errors (5.1) for every combination. We computed both the back-projection error (geometry) in the landmarks (5.2) and the intensity error in the pixels (5.3), and chose the combination of views that produced the lowest pair of geometry-based and intensity-based RMS errors. An example of this is presented in Fig. 7.9 for the synthetic dataset. Notice how the worse possible combination of basis views is along the main diagonal, or in other words when the basis views are coincident. The model generally performs better (improved synthesis results) as we increase the angular distance between the basis views, up to the point where the landmarks disappear owing to self-occlusion, and we begin the transition into a different aspect.

Once an appropriate pair of basis views is selected, we then manually synthesise all the images in the training set using the ground truth landmark positions, and recover the distributions of the 10 LCV coefficients. Based on this information we may then adjust the Gaussian priors (means and standard deviations) as we did before in chapters 5 and 6, to match as closely as possible with the recovered distributions and diversity of the 10 coefficients. So for example, if we are dealing with rotation around the vertical view-axis, then coefficients a_3, a_4 and $b_{1\ldots4}$ will be constant and as a result their priors will have a very small standard deviation. On the other hand coefficients a_0, a_1 and a_3 will have a much larger standard deviation, with a range determined by the training set and centred around the value with the highest probability.

We did not wish to restrict the optimisation algorithm by initialising inside narrow boundaries around the probable values, because this would unnecessarily reduce the diversity of the populations and stall the progress of the algorithm prematurely. In addition, such an initialisation would most likely have caused the optimisation to find a value inside the boundaries discovered during training and, considering that the test data is not inside the training set, this is obviously the wrong choice. Instead, we allowed for larger boundaries covering the whole domain inside which the 10 coefficients where defined for increased diversity, while regularising and localising the search using the priors.

Finally, by training the models in such a way, we were able to determine the range of values for the cross correlation and back-projection errors of the synthesis. This information can subsequently be



(a) Geometric error (b) Pixel intensity error

Figure 7.9: Example of basis views training errors for the synthetic dataset.

used during the validation stage in order to chose appropriate thresholds and enable us to be in a better position to judge numerically whether a particular test-run has converged successfully or not.

7.3 Proposed experiments

In this section we discuss in more detail the experiments we carried out on the three image datasets and the subsequent analysis of the results. The main theme of this thesis is the study of object recognition under changes in viewing angle, so we primarily focused our attention on pose variations in the datasets. We did however, experiment with a limited range of expression and illumination variations and the existence of occluding foreground objects and noise.

Thus, for each database we evaluated the performance of each model and the optimisation algorithm by their ability to reconstruct a given scene or target image. In every test we tried to minimise the dissimilarity between the model and the scene image by using the sum of squared differences error metric and appropriately varying the 10 LCV coefficients. The quality of the synthesis and the match between model and scene image was evaluated in the end by computing two separate metrics: the cross correlation coefficient and the back-projection error between the positions of the landmark points in the scene image and the points reconstructed by the model. In this way, we wanted to capture both the pure geometric reconstruction quality and the combined geometric and photometric synthesis in order to avoid admitting trivial solutions with high cross-correlation as correct solutions. A correct solution was chosen as the one that had higher cross-correlation and lower back-projection error values than the chosen thresholds, based on the identified error ranges during the training of the models.

In most of our experiments we used k-fold cross validation [Kohavi (1995)] as a way of partitioning each dataset and testing the model. In addition, every experiment was executed for 100 separate test runs, and the median value was returned as the accepted result. This was done in order to minimise any unusual behaviour of the optimisation algorithm and instead recover the average optimisation trend relative to the specific model-dataset combination. For the minimisation of the SSD error, in all the tests and datasets, we used a hybrid approach similar to our findings from chapter 6. More specifically, we used SOMA for a fixed number of 15000 (function evaluations) FEs and then switched over to the variable step restarting

simplex algorithm for an additional 2000 FEs.

7.3.1 K-fold cross validation

Cross-validation in general, is a method of dividing a dataset into complementary subsets and using a subset as the *training set*, while retaining the other as *testing set* for validation purposes. k-fold validation divides the data into k, mutually exclusive subsets (the folds). Each time, one of the k subsets is used as the testing set and the other k-1 subsets are combined to form the training set. The average error across all the k tests is computed. The advantage of using the k-fold method is that it is not so important how the data is divided, since every data object gets to be in a test exactly once, and gets to be in a training set k-1 times. As a result, the variance of the resulting estimate is reduced as the number of folds is increased.

Since each database is structured differently, the division of data into folds is performed in a different way in each case, and is described in the following sections.

7.3.2 Experiments on database 1 (Synthetic database)

Database 1 contains in total 301 data samples of pose, expression, occlusion and noise variation. From those we used a smaller dataset, which itself was split into secondary subsets (folds) using k-fold cross validation as follows:

- Pose variation: 11 folds, each containing the images captured from a particular view at 5^0 intervals between $\pm 25^0$, and in the same natural expression.
- Noise:
  - As above, but each scene image now contains, random Gaussian noise with σ=0.05.
  - As in the pose variation experiments, but each scene image now contains, unmodelled, random Gaussian noise with σ=0.1.
- Occlusion:
  - As in the pose variation, but each scene image now contains an occluding surface equal to 20% of the object's area randomly placed in front of the object of interest.
  - As in the pose variation, but each scene image now contains an occluding surface equal to 40% of the object's area randomly placed in front of the object of interest.
- Expression variation:
  - As in the pose variation, but each scene image has a different unmodelled, expression (happy).
  - As in pose variation, but each scene image has a different unmodelled, expression (angry).
- Horizontal-axis pose variation: 5 folds, each containing the images captured from the frontal view and at the same natural expression, but at various rotation angles about the horizontal axis (5^0 intervals between $\pm 10^0$).

We should mention here that unlike the synthetic dataset, the remaining databases contain many different individuals or objects, and therefore division of the data into k mutual exclusive folds for training and testing purposes is not possible, unless we are dealing with pose variations one object at a time. This is because an LCV model that has been trained on a specific object cannot be generalised to a new object, using the same choice of basis views (i.e. a model of a duck cannot synthesise a scene image of a car no matter how much we vary the LCV coefficients). Such a feat would only be possible if we were to consider the basis views not to be part of the modelling stage and for the purpose of testing the system we were to assume that all the basis views combinations are always known for every object, and we simply perform the training on the prior distributions of the coefficients for each model. Alternatively, a more practical way to proceed would be to allow the testing set to be part of the training set,which would be appropriate if a general description of the object has been seen before and is familiar to the system, and also if we keep a database of trained models. In this way, we may claim ignorance about the specific configuration of each object, and still obtain a meaningful optimisation outcome during testing. In more detail, we can test for false positive and false negative results and ensure that a given model of an object matches well only with an image of itself and not with images of another, different object. Therefore, keeping the latter in mind, we carried out the experiments described in the next sections.

7.3.3 Experiments on database 2 (COIL-20 database)

Database 2 (COIL-20) contains 1440 images of 20 different objects in 72 poses. From these, 5 objects were rotation-invariant owing to their specific shape and texture, and could not be easily modelled by the LCV system (see Fig. 7.4). For the remaining 15. we used 11 poses at 5^0 intervals between $\pm 25^0$ around the frontal view of 0^0. The experiments we carried out on those pose samples were:

- Pose variation: For each of the 15 modelled objects, we generated 11 folds each containing the image of that object captured from a particular view at 5^0 intervals between $\pm 25^0$.

- Object identification: We used all of the above 15 trained models, and attempted to identify the frontal view (0^0) amongst the 20 objects in the database. This resulted in the generation of a 15$\times$20 array of model$\times$object that determines the robustness of each model in terms of true/false positives and negatives.

7.3.4 Experiments on database 3 (Yale face database B)

Database 3 (Yale face database B) contains 5760 images of 10 individuals across pose and lighting variations. We carried out the following experiments on the full set of pose images:

- Pose variation: For each of the 10 individuals, we generated 9 folds each containing the image of that individual at the angles already mentioned in section 7.1.3.

- Object identification: For each of the 10 modelled individuals, we attempted to identify the frontal view amongst all the faces in the database. This resulted in a 10$\times$10 comparison array containing possible matches between model and object.

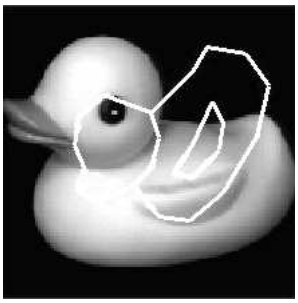


Figure 7.10: COIL-20 sample with superimposed AAM in typical starting position.

- Intensity variation: For one chosen individual (in this case subject B01), we trained a frontal view model at "neutral" lighting conditions (e.g. elevation 0 and azimuth 0) and tested the recognition rates, for the same individual, in the same frontal pose and across all 64 illumination samples.

7.3.5 Comparison with AAMs

In addition to the LCV method, we carried out the same experiments using the AAMs, a technique also aimed at solving, among other things, the pose-invariant object recognition problem. The rationale behind this is that, by contrasting our test results against a tried-and-tested technique such as AAMs, used on the same publicly available datasets, we will be able to compare our method indirectly, with many other approaches that have also used AAMs as a measure of their effectiveness and robustness. In order to aid direct comparison with the LCV approach, all the tests carried out were the same across the two methods, and we constructed AAMs (at least the shape model part) using the same sets of landmark points as used for the LCV. The only difference was in the optimisation solutions employed by the two methods. Whereas the LCV method uses a hybrid search step (as explained in section 6.3.3), the AAM uses essentially is a local search step, which can easily get stuck in false minima. The pyramid search may help avoid initial such minima, but cannot compare with the performance of global or local-restarting methods.

It is therefore necessary to ensure that a good initialisation is always available to the appearance model to avoid premature convergence. Thus, for all the AAM tests carried out, we used the following initialisation; The model was placed on a random position in the image, always overlapping (partially or totally) the scene object, using the mean trained shape, and within some arbitrary scale and rotation factors (see Fig. 7.10). In addition, each search was allowed to execute at 4 different resolution levels and at 50 (function evaluations) FEs per level, yielding a total of 200 FEs. Even though 200 FEs of a local method combined with good initialisation cannot compare with the 17000 FEs of the global search for which we allowed the LCV model to run, we would like to emphasise here that the purpose of the comparison with AAMs was not to evaluate the LCV in terms of its convergence abilities, but instead to examine how well each method can model shape (and to some extent grey-scale) variations. This is in fact denoted by the minimum error achieved in our tests, and not how many times that minimum was reached (even though that information is also reported in our results) since we are dealing with multiple test runs for each model.

So, as in the case of LCV we are presenting a complete recognition approach and we are interested in both the quality of the minimum and (the probability of) its occurrence, in the AAM case we are only aiming to compare with the minimum reached in the LCV. The focus on this comparison is mainly owing to the fact that the appearance model may be able to capture combined appearance variations better than the LCV (although the latter does not make any claims about accurate grey-scale variation, only shape and pose) and it would be interesting from a theoretical point of view to compare the capabilities of each model. After all, if consistent performance is required from the AAM (at the expense of fast convergence) it is quite possible to replace the model search-and-update step (algorithm 6) with another global method (see Chapter 6). Furthermore, the AAMs are a well-known and widely used technique for pose, shape and appearance variation that constitute something of a baseline, against which every equivalent method can be compared. The accuracy and efficiency numbers we are quoting here for both the LCV and AAMs (in its current implementation) are not intended to determine which of the two methods is better (since we are not dealing with similar optimisation approaches) but to show how much more (and if at all) our solution with the addition of a Bayesian model and hybrid optimisation method, improves over this widely-used standard.

7.4 Results

This section presents the comprehensive results of the aforementioned experiments on the three datasets. We begin with the pose variation for all the databases, followed by noise, occlusion, expression and horizontal pose variation for the synthetic database. In addition, we present some limited data on illumination changes from the Yale B face database. In all our data, we quote the cross-correlation coefficient and back-projection errors between the target scene and synthesised images, and include the results from the AAM test runs on he same datasets for ease of comparison. Any overall conclusions on the performance of LCV on successively more complex data, as well as how it compares with AAMs, is given in the summary section of this chapter.

7.4.1 Database 1

The first set of results for the pose variation in the synthetic database are summarised in Fig. 7.11, which compares the two errors: root mean square error (RMSE) and mean absolute error (MAE). The MAE is a quantity that is used to measure how close predictions are to eventual outcomes. In other words, it measures the magnitude of the errors in a set of forecasts without considering their directions. Since the MAE is a linear score, all the residuals are weighted equally in the average. The RMSE uses a quadratic scoring rule and thus the errors are squared before they are averaged. For that reason, the RMSE gives a relatively high weight to large errors, and is most useful when such residuals are particularly undesirable.

The MAE and RMSE may be used together to diagnose the variation in the errors in a set of forecasts. The RMSE will always be larger or equal to the MAE and the greater the difference between them, the greater the variance in the individual errors in the sample. If RMSE is equal to the MAE then the individual errors in the sample are of the same magnitude. Both the RMSE and MAE range from $[0, \infty)$. In Fig. 7.11 we see that both errors are quite low and of approximately similar magnitude, since

the difference between the RMSE and MAE is small and ranges between 0.01 and 0.04. Note that both RMSE and MAE were calculated using the cross-correlation between the observations (test runs) and the ground truth. In addition, we see that the errors remain quite stable throughout the range of pose angles (between $\pm 25^0$) indicating that we have an equally good chance of reaching the identified ground truth values, independent of pose. This graph however does not tell us the values of the cross-correlation reached, just how close we arrived to the ground truth.

To explore the former, we need to look at Fig. 7.12. This figure shows three different pieces of information. First is the average CC (bold line) calculated as the mode of the sample for different pose angles. We chose the mode instead of the mean, because we were interested in the solution which had the highest probability of occurring. This makes more sense from an object recognition point of view, rather than the mean value, which is affected by outliers and does not really say much about the recognition accuracy of the algorithm. A good optimisation algorithm is one that recovers an acceptable solution the majority of times. In addition, the graph shows the average ground truth error (dashed line) and the empirical threshold error (solid line). The first is the CC error that was identified by solving the system of linear equations (3.14) given the ground truth scene and the correct landmark positions on the object. The second, is the minimum CC error that was empirically discovered by looking at each of the 100 test results, for every pose angle, and deciding based on purely qualitative criteria whether the synthesised image was a good representation of the shape, pose and intensity of the scene view, similar to the experiments in section 5.3.

By close observation of the CC error, we see that the most common results are considerably above the empirical cut-off line (below which would most likely indicate a pose recognition failure) and also higher than the ground truth error, for pose angles $-20^0, ..., 25^0$. This is perfectly possible and acceptable, since the ever-improving effect of the optimisation algorithm on the objective function can reduce any minor inconsistencies in the landmarks or the approximation errors in the pixel values. Only for -25^0 do we see that the CC error is lower than the ground truth, but still well above the empirical threshold. Also, notice the characteristic slight falloff at the farthest angles $\pm 20^0$ and $\pm 25^0$.

Another graph that supplements the average CC information, is the histogram that incorporates all the results from the 11×100 test runs (Fig. 7.13). Here we can identify the mode of the sample and how close it lies to the mean ground truth and empirical errors respectively (horizontal lines). It is immediately obvious that the histogram is unimodal with a well defined peak at 0.9875 well beyond the thresholds (ground truth (g.t.)=0.9728, empirical=0.9513), and with few insignificant outliers[1] that fall of sharply as the cross-correlation score gets lower.

All the above graphs were related to the cross-correlation error, which combines both geometric and photometric information. As we already know, the latter may overpower the former and yield a high CC solution that might not be geometrically accurate. This is why we calculated the back-projection (BP) error in the landmarks, a pure geometrical measure, and generated similar graphs. We start with the average BP graph (again with the mode of the sample) for different pose angles (Fig. 7.14). Here we

[1] In terms of containing other significant modes that would signify the presence of local minima. Although outliers exist we are confident that they are the result of the occasional failure of the optimisation algorithm to converge.

see a similar pattern, with lower BP errors for frontal and near-frontal angles and with the characteristic, gradual falloff for angles over $\pm 20^0$. The mode again is below the empirical threshold (also demonstrated by the BP histogram in Fig. 7.15). However, this time the ground truth BP error is much lower than any solution recovered. This may be explained in part by the fact that the optimisation algorithm operates on the combined cross-correlation error and may sometimes sacrifice geometrical accuracy for an improvement in appearance. Also, it may be argued that between BP error values of different magnitudes, there may not necessarily exist a qualitative difference of equal magnitude, on the synthesis of a novel view. It may for example be possible that a single outlying landmark affects the overall BP error (since its an averaged value), even though the results are identical to the viewer. We therefore point out that despite the fact that the mode of the BP error is higher than the ground truth, it still represents a very good and acceptable solution. This is exactly the reason why we chose to use the additional empirical, qualitative threshold and consider both the BP and cross-correlation errors in tandem.

The next figure (Fig. 7.16) shows the diversity of the mean coefficients from all the test runs for every pose angle. We immediately see a pattern similar to Fig. 5.3, where coefficients b_j are stationary at their optimal values (since there is no horizontal axis rotation) aided by the narrow Gaussian priors. From the a_i coefficients (responsible for vertical axis rotation), a_2 and a_4 are centred at zero with no diversity and a_1, a_3 range from approximately -0.5 to 1.5 for rotation angles $\pm 25^0$. Coefficient a_0 which varies with object translation and is of different units than the rest of the coefficients, has a much larger diversity, as is usually expected (see section 5.1.4). The diversity graph helps to establish how well the recovered coefficient range captures the underlying transformation (in this case very well), and if there are any outliers outside this expected range (not any significant outliers here). Such outliers may be the result of failed optimisation attempts or the existence of an important locally optimum solution.

The final two graphs we present for the pose variation, are the colour-map plots Fig. 7.17(a) and (b) for the cross-correlation and BP errors respectively. Their purpose is to illustrate the acceptance percentage (i.e. how many test runs were below or above the given empirical threshold value) for all the test runs (not only the mode of the sample) at different threshold levels, in order to get an idea about the overall efficiency of the optimisation algorithm for this particular dataset. The colour-map plots are essentially 3-dimensional and depict acceptance percentage (grayscale colour) as a function of threshold and pose. The empirical threshold lines are also included. We see that in general for the CC (Fig. 7.17(a)) the acceptance ratio above the empirical threshold is in the range of 50-70%, increasing for frontal and near-frontal angles. Note again that these graphs illustrate the average efficiency of the algorithm and not the accuracy, since the latter is captured by the mode of the sample we have seen previously. A very probable, good result that lies on the peak of the histogram makes the optimisation algorithm very accurate, but if at the same time we obtain many outliers (thus reduced acceptance percentage) the algorithm is ineficient. We see a similar pattern for the BP error thresholds, with the acceptance percentage in the mid-50s to 70s as we cross the empirical threshold. The same results are also summarised in Table 7.1, for the empirical thresholds for each pose, and the average, overalll acceptance score. The final column of Table 7.1 only admits solutions that are within both the cross-correlation and BP empirical

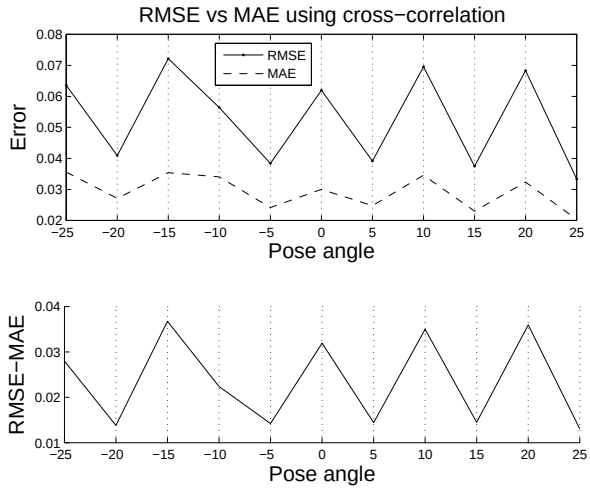


Figure 7.11: RMSE and MAE plots using cross-correlation.

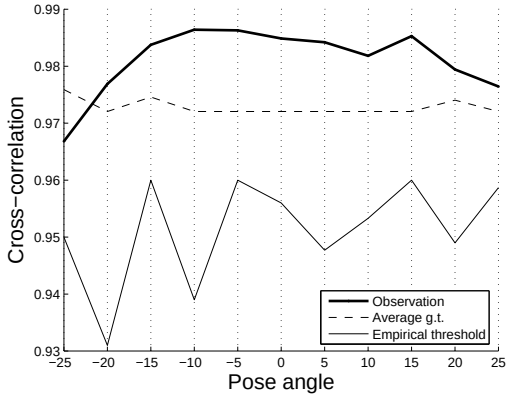


Figure 7.12: Average cross-correlation plot (mode of sample).

error thresholds. The average row at the bottom represents the portion of the histograms (Fig. 7.13 and Fig. 7.15) that are on the left or on the right of the empirical threshold horizontal lines respectively.

We can now compare the above results with those from the AAM tests on the same dataset. Fig. 7.18 shows that AAMs perform very well between the angles $\pm 20^0$, but less so in the more distant angles at $\pm 25^0$. This may be attributed to a possible inability of the AAM to accurately extrapolate data, since between the angles $\pm 20^0$ the missing information is interpolated. Although the AAMs find solutions well above the empirical thresholds, they still cannot match the results of the LCV approach, as far as cross-correlation is concerned. The same graph, but using the landmarks back-projection error (Fig. 7.19), reveals a somewhat different picture, and shows the AAM outperforming the LCV for certain angles, although not to a great extent. The former does however have a much better degree of accuracy in the angles $\pm\{20^0, 25^0\}$

If we examine the RMSE vs MAE graph (Fig. 7.20), we can see that both the RMSE and MAE errors are larger than in the LCV case (Fig. 7.11), but this is a direct result of the lower CC values recovered by the AAM. It is also apparent that the magnitude of the errors between the two measures is

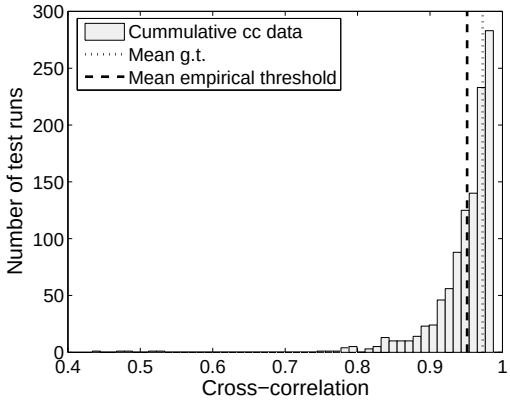


Figure 7.13: Full cross-correlation data histogram for the pose variation.

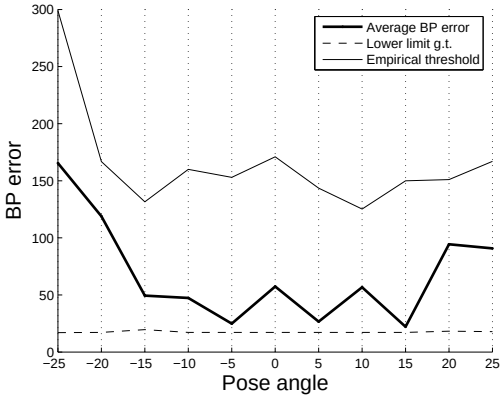


Figure 7.14: Average back projection error (mode of sample).

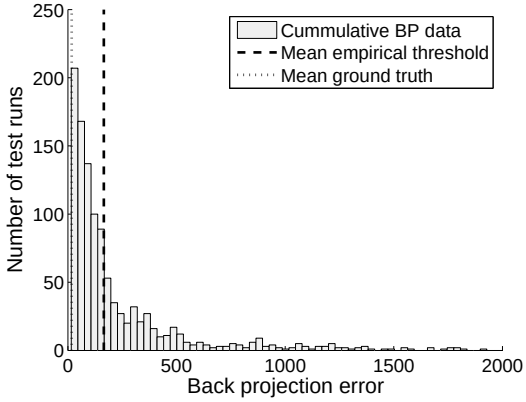


Figure 7.15: Full back projection data histogram for the pose variation.

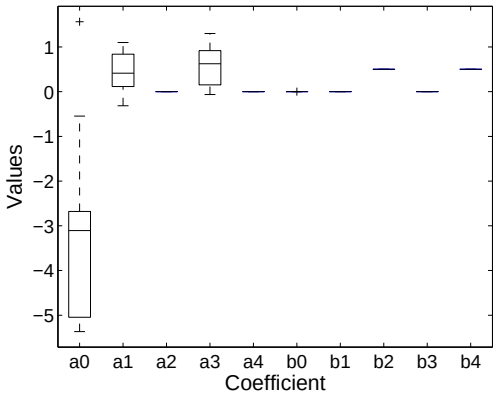
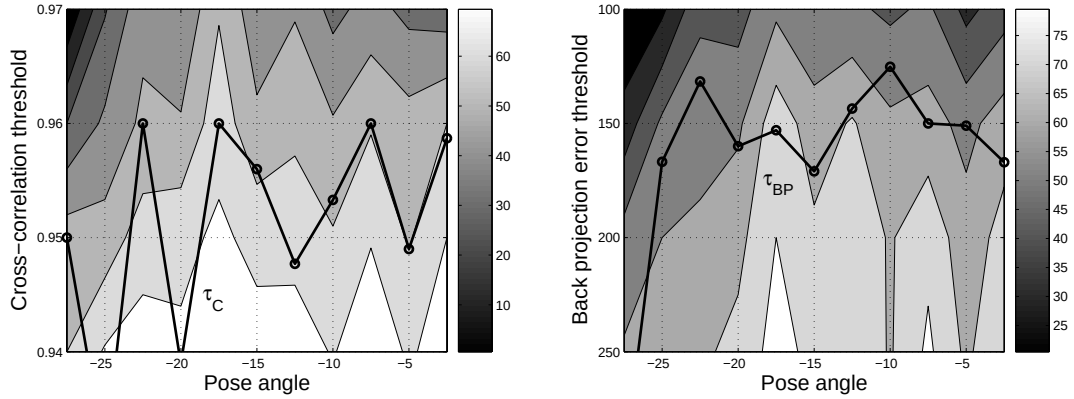


Figure 7.16: Diversity of mean coefficients for pose variation.



(a) Cross-correlation (b) Back projection

Figure 7.17: Acceptance % of test results for different thresholds.

Pose^0	Empirical CC	Empirical BP	Empirical c.c + BP
-25	55%	71%	54%
-20	80%	56%	53%
-15	52%	55%	49%
-10	74%	65%	63%
-5	66%	77%	65%
0	57%	67%	56%
5	70%	67%	67%
10	59%	57%	57%
15	59%	64%	59%
20	61%	57%	53%
25	64%	70%	73%
Average	63%	63%	57%

Table 7.1: Acceptance results for pose variation at different thresholds.

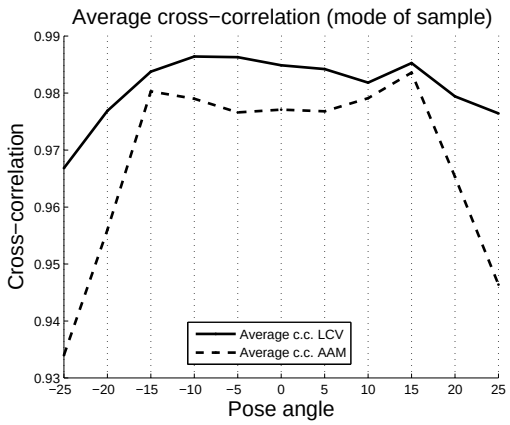


Figure 7.18: Average cross-correlation plot (mode of sample) using AAMs.

more stable than the LCV case. Such an outcome indicates that although the distance of the residuals from the mean ground truth is higher than before, there is now much less variance in them between different pose angles. We can also see this if we examine the data in Table 7.2. As before, the table shows the recognition results (acceptance percentage) for all the test runs at various CC and BP thresholds. The difference in this case however, is that there is almost no variance between subsequent threshold values and between using an intensity or a geometric based threshold. This is due to the local optimisation algorithm used, which in reality offers two possibilities: either convergence very close to the correct solution, or convergence very far away and/or collapse to a single point. It may still be possible to become trapped inside some nearby local optimum if such one exists. This would indicate a likely problem with the objective function formulation, that should be addressed, and not with the optimisation algorithm itself. However, we did not encounter any such optima, as demonstrated by the results in Table 7.2.

In conclusion we can say that the LCV method is more accurate for pose recognition, in this particular dataset, especially at frontal/near-frontal angles and when the CC score is considered. However, on average the AAMs are more efficient at the empirical thresholds chosen, provided a good initialisation is available for the optimisation search. Both methods perform well in finding the global minimum at close proximity to the known ground truth. It remains however to see how well this can scale to more complicated datasets and the existence of noise, occlusion and localised expression changes.

7.4.2 Database 2

The next set we will consider is the COIL-20 database, which is more demanding than the synthetic set since it contains real-image data under realistic illumination conditions, but at the same time we are still searching over a constant background. This should help maintain the optimisation process within manageable limits.

We first consider the object identification results, in which 15 models are compared against the full 20 objects in the frontal view. The goal here is to evaluate the performance of each model in the presence of unknown classes of objects to the system. Throughout the various, resulting 15×20 model$\times$object

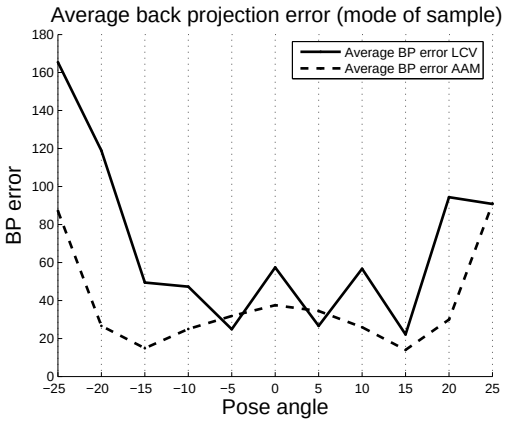


Figure 7.19: Average back projection error (mode of sample) using AAMs.

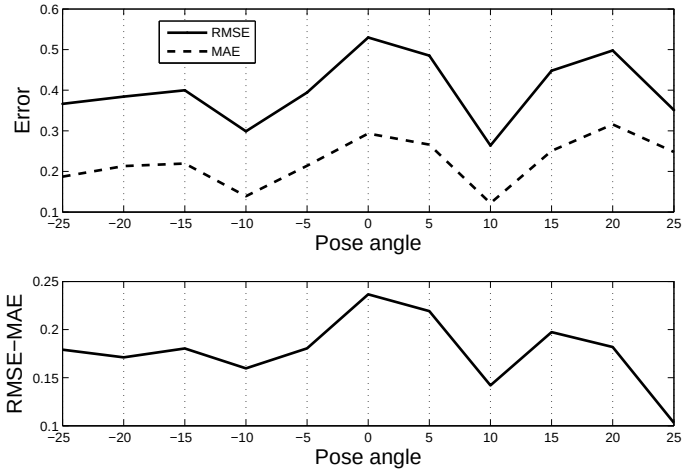


Figure 7.20: RMSE and MAE plots for cross-correlation, using AAMs.

Pose[0]	τ_{cc_1}	τ_{cc_2}	τ_{cc_3}	τ_{cc_4}	Emp. CC	τ_{BP_1}	τ_{BP_2}	τ_{BP_3}	τ_{BP_4}	Emp. BP	Both
-	0.95	0.96	0.97	0.98	-	200	150	100	50	-	-
-25	0	0	0	0	0	80	80	80	0	80	80
-20	72	0	0	0	72	72	72	72	72	72	72
-15	74	74	74	74	74	74	74	74	74	74	74
-10	89	89	89	0	89	89	89	89	89	89	89
-5	70	70	70	0	70	70	70	70	70	70	70
0	68	68	68	0	68	68	68	68	68	68	68
5	76	76	76	0	76	76	76	76	76	76	76
10	82	82	82	0	82	82	82	82	82	82	82
15	77	77	77	77	77	77	77	77	77	77	77
20	62	62	0	0	62	62	62	62	62	62	62
25	0	0	0	0	0	58	58	58	0	58	0
Avg.	60.9	54.3	48.7	13.7	60.9	73.4	73.4	73.4	60.9	73.4	60.9

Table 7.2: Acceptance results for pose variation at different thresholds, using AAMs.

arrays, we would like to observe a clearly defined error response, located approximately[2] in the main diagonal, with possibly high recognition rates for when model=object, and low or zero false positives and negatives.

In the same way as before, we first consider the RMS error of each of the 100 test runs from the average CC ground truth. The results from all the object identification tests are combined into the greyscale plot array in Fig. 7.21. In general, we see a well defined diagonal (darkest colour) with low RMSE response ranging from 0.015$\sim$0.045. This response is shown in the bottom sub-figure where the minimum RMSE values in each row (corresponding to each model) are plotted. We note however that there are a few inconsistencies for models 7, 8 and 11, in that they produce a lower RMSE response at objects 8, 16 and 16 respectively. This is shown as a deviation from the approximate main diagonal and is illustrated with the overlaid white line that connects the minimum RMSE values from each row. Not however that the apparent deviation for models 13 and 19 is quite normal since the models are not numbered sequentially after model 11 on the ordinate.

The RMS error of course is not used as a measure of the recognition accuracy in this case, since it reports the average distance from the mean ground truth, whereas we are more interested in the most likely (probable) CC response in all the 100 test runs. Nevertheless, the RMSE can serve as a good indicator of the combined accuracy and efficiency performance of each model. We therefore expect to get good overall accuracy (CC scores) and efficiency (acceptance ratios) for the 15 models, except perhaps in the case of models 7, 8, and 11 where me might have lower associated acceptance scores.

Furthermore, by close examination of the two objects which cause the false positive responses, 8 and 16, but also the adjacent object 7, we see three vertical areas of high RMS error that cover most of the models in their respective columns. Such an observation implies the existence of objects are of generic enough shape and texture that can easily match most models given an appropriate transformation. On the other hand, objects that match well only to their respective models appear as light-coloured columns, in this example, objects 3, 6, 18 and 19. We expect to see these results mirrored in the cross-correlation colourmap array.

This is indeed the case for Fig. 7.22. Objects 7, 8 and 16 still produce the familiar high correlation responses, however they are not large enough to cause a mismatch between model m_i and object o_j when $i \neq j$. This becomes more apparent if we examine the line that connects the highest CC score in each row, with fits perfectly to the model=object diagonal and with the absence of any outliers. Furthermore, the sub-figure shows a comparison plot between the minimum of each row above (coinciding with the diagonal line) and the ground truth error. In there we see that the observation line (mode of test run data) is above or very near at the g.t. threshold line. These two plots therefore provide a good indication that when the CC measure is used, we have perfect classification results across different objects for every model and with very high accuracy. What remains to be seen is the efficiency and the results that we get when we consider a geometry-only matching score such as the BP error.

The efficiency can be seen in surface plot form in Fig. 7.23. In this plot, there is a very high recogni-

[2] Approximately, since not all the objects are modelled due to the rotation-invariant properties in some.

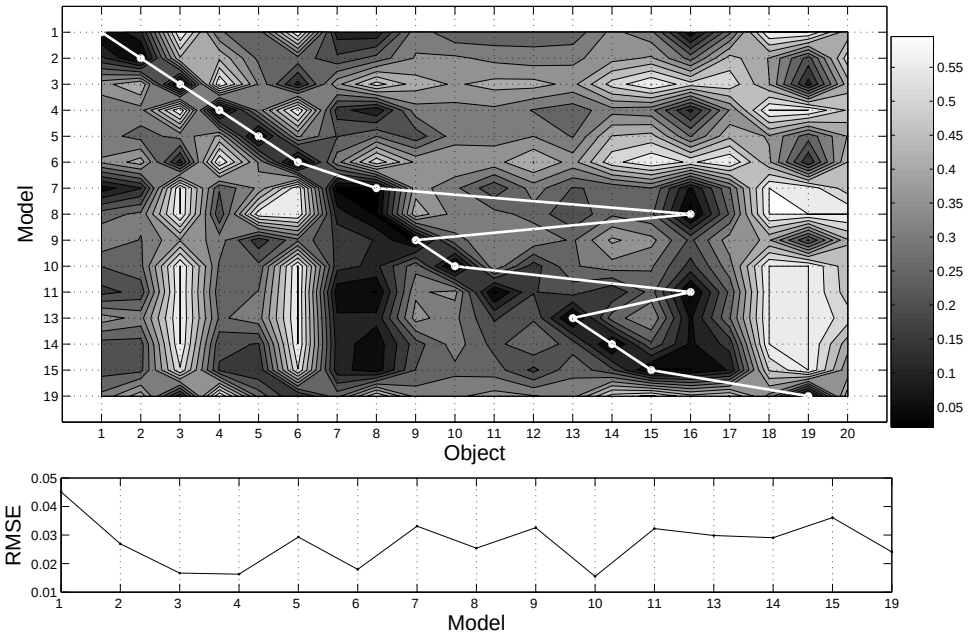


Figure 7.21: RMSE model×object array for the frontal pose using LCV.

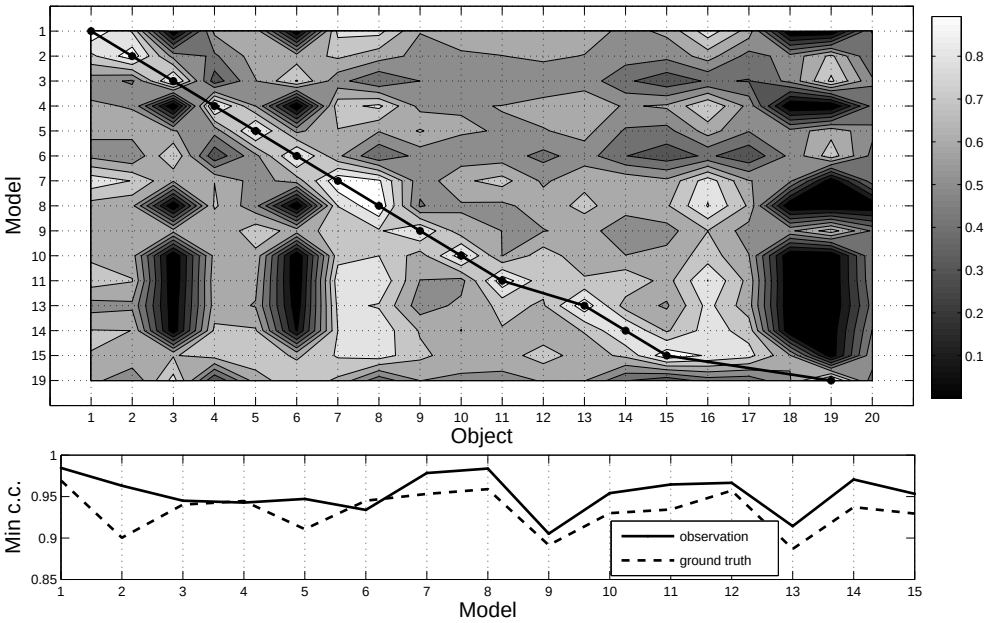


Figure 7.22: CC model×object array for the frontal pose using LCV.

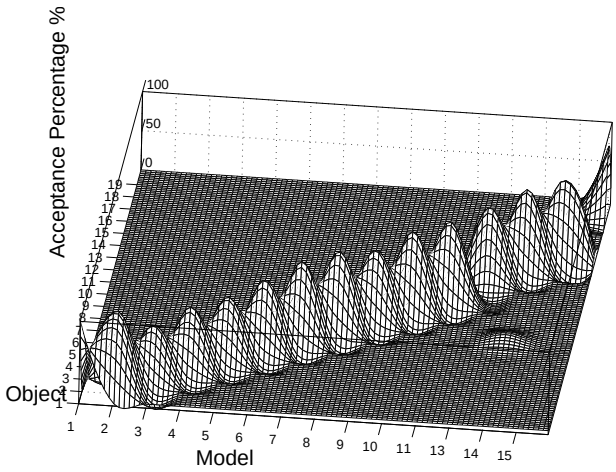


Figure 7.23: Acceptance ratio for Model×object at the frontal pose using cc.

tion percentage response for when model=object, ranging from 80% to almost 100%. Conversely, when model≠object, we obtain a flat surface at 0% recognition. Only one occurence at approximately 10% for model=14 and object=8 is visible, but that is too small and insignificant to cause any misidentifications. Such a limited and localised responce could be due to a non-optimal setting of the empirical threshold, which as we have pointed out is a manual and subjective process and therefore not exact. Alternatively, it may be due to a difference in CC levels between various model-object (m_i, o_j) combinations. For example, for (m_1,o_1) we might get a score of c_1 and c_2 for (m_2,o_2). If c_1 is much lower than c_2 it is possible that when (m_1,o_2) to get a score c_3 where $c_1 < c_3 < c_2$ that is eroneously interpreted as a successful match.

If we examine the same graph but this time using the BP error (Fig. 7.24), we note that this small inconsistency has now disappeared. This is because the geometrical BP error is less likely to produce such mismatches, which usually occur during optimisation with the CC, a process that is known to be able to compensate by adjusting the shape of the model (usually a 2-D affine transform) so that the overall appearance produces a false, positive match.

Finally, for the LCV object identification experiments on the COIL-20 database, we present the 15×15 model×object array using the BP error in the landmarks (Fig. 7.25). It is only 15×15 since 15 object are modelled and thus just 15 out of 20 have associated landmarks we can use to calculate the BP error. Additionally, in order to preserve the detail in the grayscale plot for low BP values (the portion of the data we are most interested in) we have set an upper limit at 1000, so any BP scores above that threshold were capped accordingly and appear as constant, white areas in the graph. Furthermore, since we are dealing with different objects, with varying geometries and thus number of allocated landmark points, it is necessary to define a strategy for calculating the BP error at each model-object combination. We have decided on an approach which attempt to equalise the two shapes by removing the most remote landmarks from the object with the largest number of points. If the number of landmarks between the model and object is the same, then we can proceed as normal

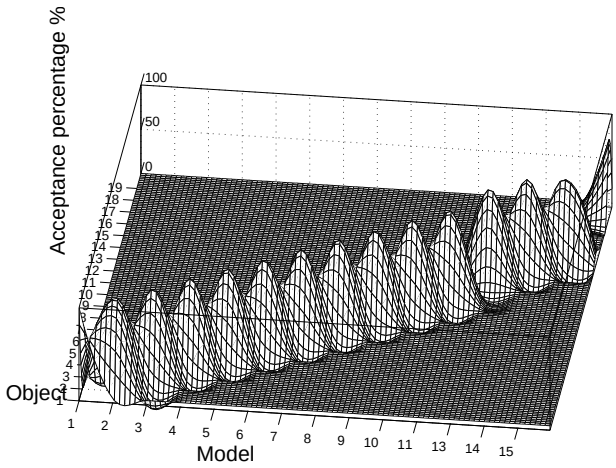


Figure 7.24: Acceptance ratio for Model×object at the frontal pose using BP.

and calculate the BP error. If on the other hand it is different, we first determine if the synthesised view $L_1 = \{p'_1, ..., p'_n\}$, or the scene view $L_2 = \{p_1, ..., p_k\}$ contain the most landmarks, i.e. $n >$ or $< k$ and: for each point in the larger dataset calculate the distance from every point in the other set $D = \Big\{\{d_{p_1,p'_1}, ..., d_{p_1,p'_k}\}, ..., \{d_{p_n,p'_1}, ..., d_{p_n,p'_k}\}\Big\}$. Then only consider the minimum distance for each landmark $D' = \Big\{\min\{d_{p_1,p'_1}, ..., d_{p_1,p'_k}\}, ..., \min\{d_{p_n,p'_1}, ..., d_{p_n,p'_k}\}\Big\}$ and finally discard the landmark(s) p_x where D' is maximum i.e. $\max(D') = d_{p_x,p'_y}$. Once the number of landmarks is the same, we can calculate the BP error as before. This way we assume that the discarded landmarks are "outliers" and try to approximate the two geometries.

Even though such an approach might not be strictly correct since a synthesised object changes dramatically when a landmark and thus a triangle is removed, from a practical and geometrical point of view it is sensible and it helps to obtain a BP score for radically different objects that would otherwise be comparable only by via the combined appearance CC score.

If we keep the above points in mind and return to (Fig. 7.25) we can identify a distinct main diagonal of low geometrical error whose value is very close to that for the ground truth as demonstrated by the sub-plot. There are also two interesting observations in this figure that we would like to analyse further before we proceed on to the AAM portion of the tests. First, we see that the observed objects of "generic appearance" that tend to match with most models from Fig. 7.22 have now shifted from 7, 8 and 16 to 2 and 9. Further examination of these objects reveals that object 9 (Fig. 7.4) has a rectangular shape with very few landmarks and boundaries that can stretch and rotate to fit well with a large number of models in the database. Object 2 (Fig. 7.4) on the other hand has a more complex, non-generic shape, that looks like it will be difficult to match with anything other than its own model. It does however have only 10 landmarks owing to its straight boundaries and its almost constant texture and thus will easily generate a low BP score when compared to most models even if it does not reproduce well details of their appearance.

The second observation is that for the first time we obtain results for models that can fairly adequately match to the majority of objects in the database. This is depicted as horizontal lines (rows) of

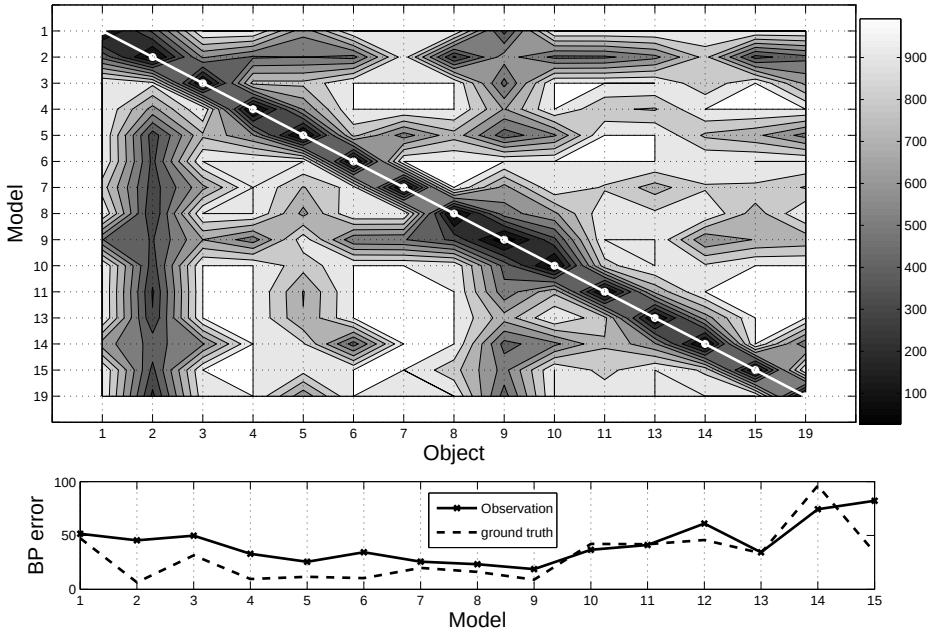


Figure 7.25: BP model×object array for the frontal pose using LCV.

medium-to-low BP error in Fig. 7.25. More specifically, this occurs for models 2, 5 and 9. We have already commented on the properties of objects 2 and 9 above and, as we can see from Fig.7.4, object 5 is geometrically very similar to object 9. In general, we would not expect it to be unusual to encounter objects that match fairly well to many models. Despite how high these mismatches may be, they do not cause deviations of the selected best match from correctly lying on the main diagonal.

The same object identification experiments have been carried out using AAMs. In this section the results are compared with the LCV approach we have analysed previously. We begin with the RMSE plot (Fig. 7.26) which when compared to Fig. 7.21 this plot reveals a generally increased RMS score ranging from 0.4 to 0.9 indicating that there is a definite drop in efficiency in this case. Additionally, we see a higher disagreement between the (white) line that connects the minimum RMS scores in each row (i.e. for each model) and the model=object approximate diagonal. This disagreement might also point to a reduction in CC accuracy, especially when it is measured against the ground truth.

To obtain a clearer view on this, it is necessary to examine the average cross-correlation response from each model×object combination in Fig. 7.27. We see that the identification accuracy remains at high levels similar to what it was when the LCV was used in Fig. 7.22. The main diagonal is clearly defined except for model=object=14 where it has failed to converge to the optimum solution. For all other models, the response is at or above the g.t. threshold. It is also the case that we have false responses when model≠object which are more distinct (a darker colour and thus representing lower CC values) than the correct matches on the main diagonal, much more so than when the LCV approach was used. This is evidence of a better separation between true positive and false negative responses that in turn helps to avoid object misidentifications. We believe the above to be a direct result of the limitations of the rather basic optimisation algorithm built into the approach using AAMs. It is only able to search

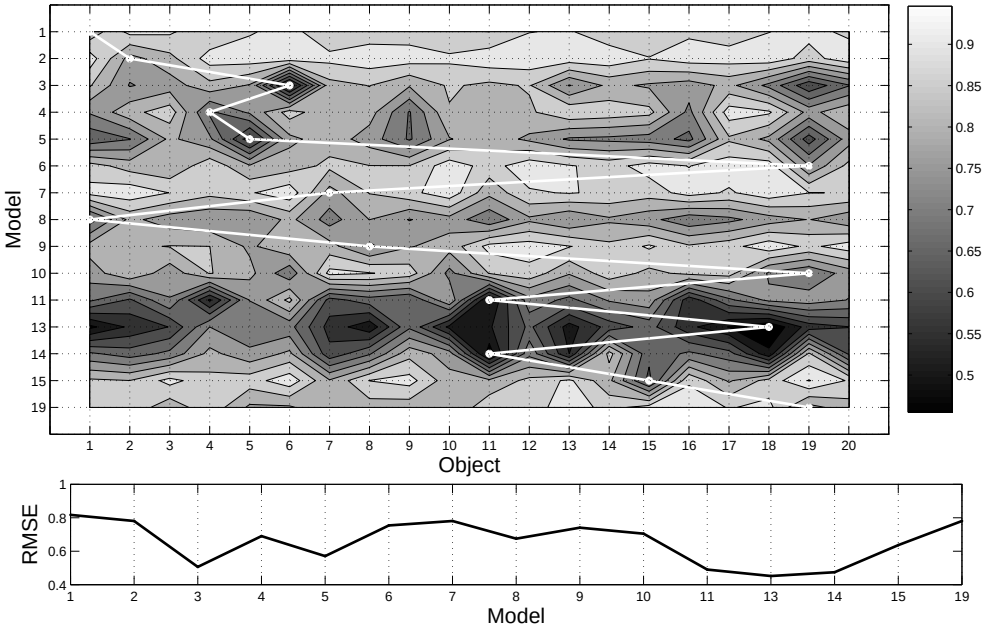


Figure 7.26: RMSE model×object array for the frontal pose using AAMs.

within a limited, local area. Thus, if it is initialised inside the basin of attraction of the correct match, it will converge to the desired correct match. If not, or if such a basin of attraction does not exist, the optimisation will get stuck fairly quickly and not attempt to recover a sub-optimal configuration with a high-enough CC response that may register as a mismatch. The hybrid method applied to the LCV approach on the other hand gives higher overall CC scores (lighter colours in Fig. 7.22) and exhibits the familiar vertical columns of high correlation.

We can now proceed to the BP error grey-scale plot (Fig.7.28) which is also limited to a maximum of 1000 in order to maintain the level of detail at the lower BP values. When compared to the results of the LCV tests the experiments with the AAMs give a lower BP error along the main diagonal (except for the non-convergence when model 14 was used) and thus have a better geometric accuracy than the LCV. This is something we have seen previously in the synthetic dataset. In the background area when model≠object and mismatches occur, we see vertical and horizontal lines resulting representing "generic objects or models" in the same places as when the LCV was used. However the BP score is now lower and as a result much closer to the optimal response on the main diagonal. This results in the plot in general appearing darker than that obtained when the LCV was used (Fig. 7.25). This improvement is in the opposite sense to the reduction in cross-correlation in the experiments we have analysed previously . We may thus conclude that even though the accuracy is slightly better for the AAM approach than it is for the LCV approach, the distinction between a correct and possibly incorrect match has been reduced.

The final investigative step into comparison with the use of AAMs for model/object identification involves examination of the optimisation efficiency, which is illustrated by the average appearance acceptance plot in Fig. 7.29. As has been the case so far, in comparison with the LCV approach, the AAM returns very low acceptance scores for the same empirical thresholds. In this particular set of tests

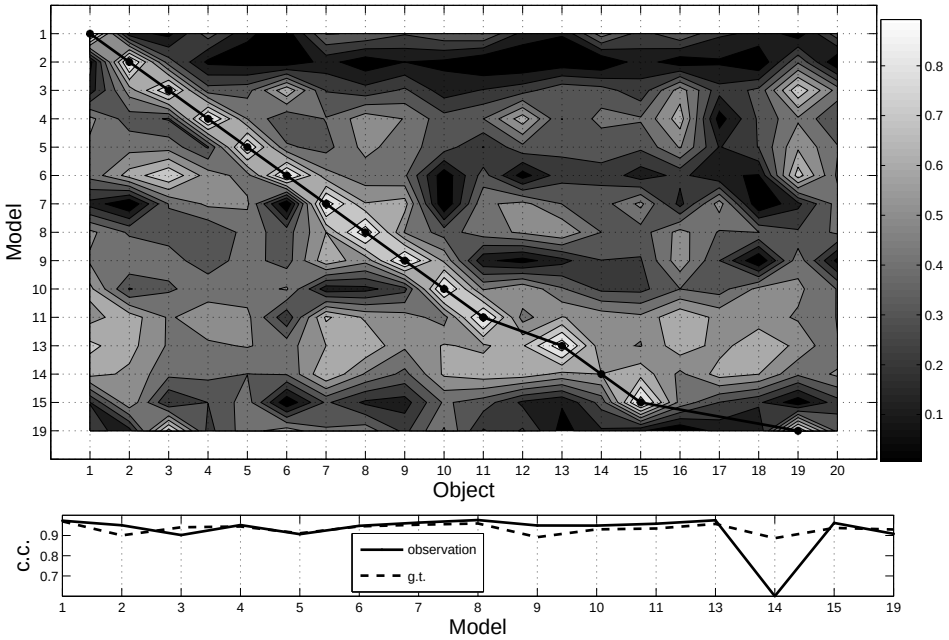


Figure 7.27: CC model×object array for the frontal pose using AAMs.

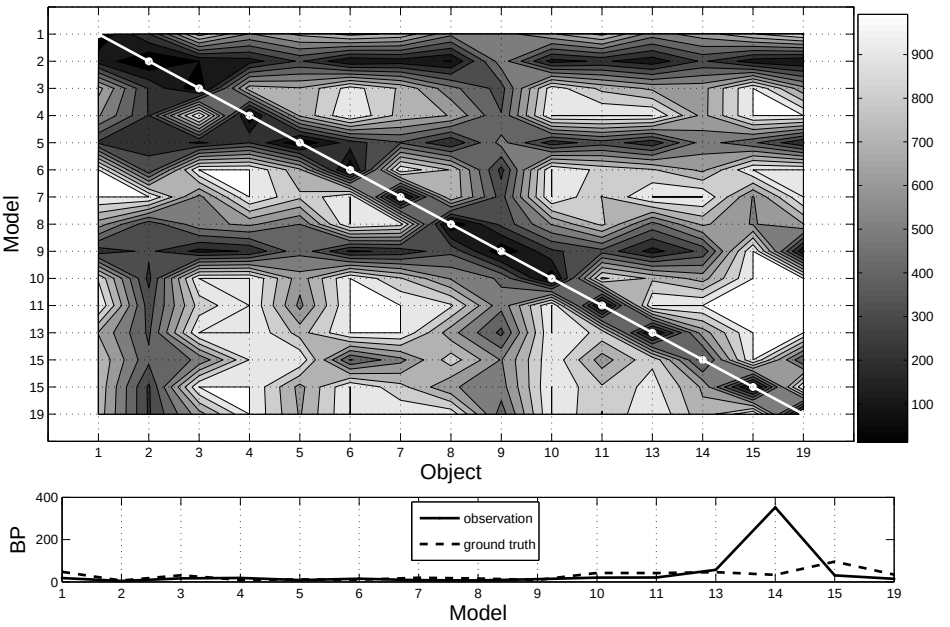


Figure 7.28: BP model×object array for the frontal pose using AAMs.

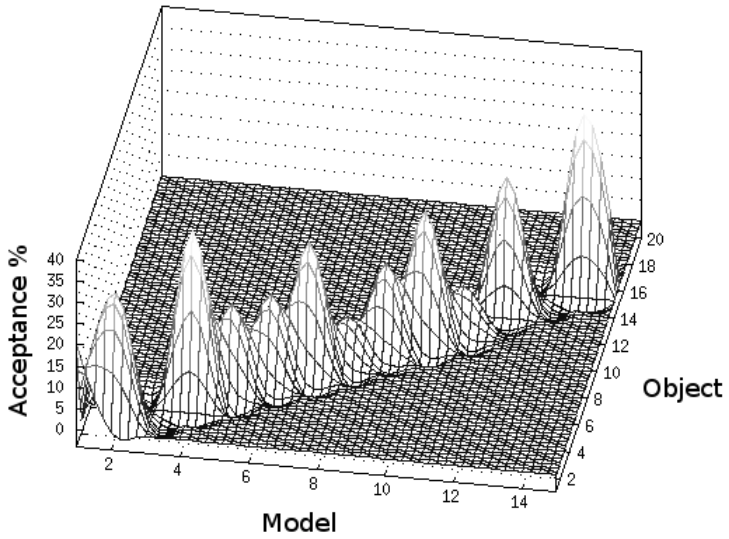


Figure 7.29: Acceptance ratio for Model$\times$object at the frontal pose using AAM.

the recognition efficiency has dropped from 80-100% to 0-40%. This again appears to be based on the inability of the local optimisation to consistently recover good solutions.

As a conclusion we would like to point out that the LCV models maintain their good performance in the presence of real-image data with high accuracy and acceptance scores. Although there are fairly high responses for some generic-looking objects in both the CC and BP measures they are not high enough to cause any mismatches, and the instance of each object is always correctly identified in each image. Compared to the AAM the LCV gives results of almost equal accuracy, but when it comes to efficiency, the local optimisation algorithm used in the AAMs (even if initialised close to the solution) cannot compare with the consistent performance of the hybrid global optimisation method built into the LCV.

However this increase in efficiency rate also brings some possibly undesirable, minor side-effects such as the discovery of sub-optimal solutions with good CC and BP scores when model$\neq$object. Although they did not pose any problems in our tests they might be a potential source of false positive matches in another scenario when the ground truth CC or BP values of the correct, positive matches are inherently not so good. If such false positive mismatches commonly occur, it is not because of a particular problem in the optimisation algorithm (if anything they are amplified by its exceptional ability to explore a large number of possible solutions) but are a property of both the model/object and of the match metric used. In other words, they are a property of the form of the objective function. Complete avoidance of this problem might not be possible and it may thus require a re-evaluation of the modelling process and of the error metric used.

The final set of tests for the COIL-20 database involve the recovery of the correct pose angle for each model when we know that the object of interest is present in the target, scene image but seen from an unknown viewpoint. We have considered views between $\pm 20^0$ at 5^0 intervals sampled in a

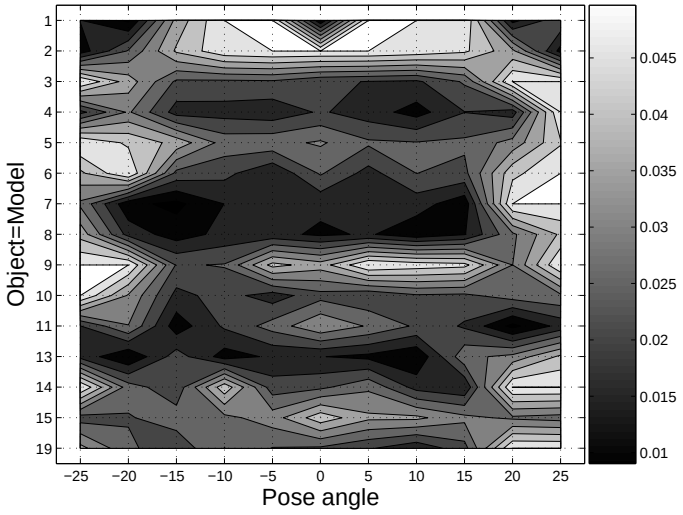


Figure 7.30: RMSE pose variation plot for the COIL-20 database, using LCV.

similar test/training set fashion as with the synthetic database discussed previously. In addition we tested against the two extreme, untrained views at $\pm 25^0$ to assess the capabilities of the models to extrapolate. The combined pose variation results are incorporated into the following 15×11 grey-scale graphs of model=object×pose.

The first graph is the RMSE plot in Fig. 7.30. We see that the scores range from 0.01 to approximately 0.05 which we consider generally very good based on previous test runs with other datasets. It is further evident from the darker patches in the graphs that pose angles between $-20^0, ..., -5^0$ and $5^0, ..., 15^0$ have the lowest RMS error values. If we look at specific objects, we can see that objects 4, 6, 7, 8 and 11 (a mixture of both generic and non-generic looking shapes) have the best agreement with their g.t. values in the majority of poses. Moreover for some objects there is a slight drop in RMS error at 0^0 which is the familiar "M" shape we have previously encountered in various 2-D pose angle plots.

Next is the CC grey-scale plot in Fig. 7.31, where we observe that the majority of scores are above 0.9 especially for the angles representing frontal views, except for object 14 for which the CC fluctuates approximately between 0.85-0.9. We also note that objects 7, 8, 13 and 15 have the highest responses for angles representing near-frontal views. Amongst this set of objects only objects 7 and 8 also have a similarly low RMS error in Fig. 7.30. This seems likely to occur either due to a high CC value in the ground truth thresholds or is due to a lower acceptance ratio for objects 13 and 15. In addition, objects 13 and 15 (the former has a non-generic shape and the latter almost looks rotation-invariant) consistently produce high CC values throughout all the poses. This of course does not mean that for all the other objects we fail to recognise the correct pose, but that for the large angles of $\pm 20^0 and \pm 25^0$ some models have more difficulty in recovering the optimal object configuration (i.e. viewing angle). As already mentioned this graph reveals an overall high CC score which should translate to a high acceptance percentage (at least for the angles which are not large) as we shall see later on.

We continue with the BP error plot (Fig. 7.32) which illustrates very good geometric matching for

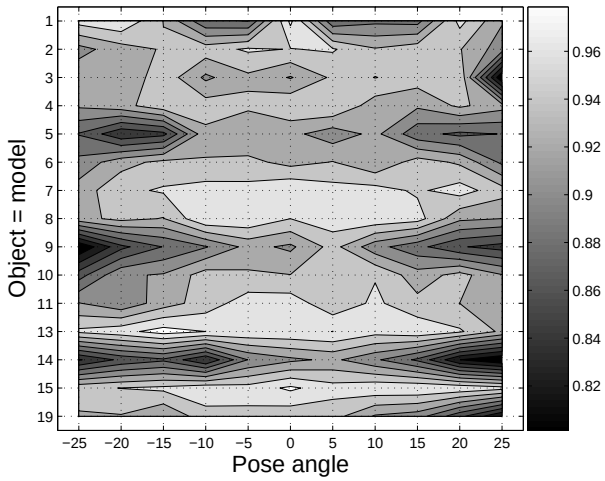


Figure 7.31: CC pose variation plot for the COIL-20 database, using LCV.

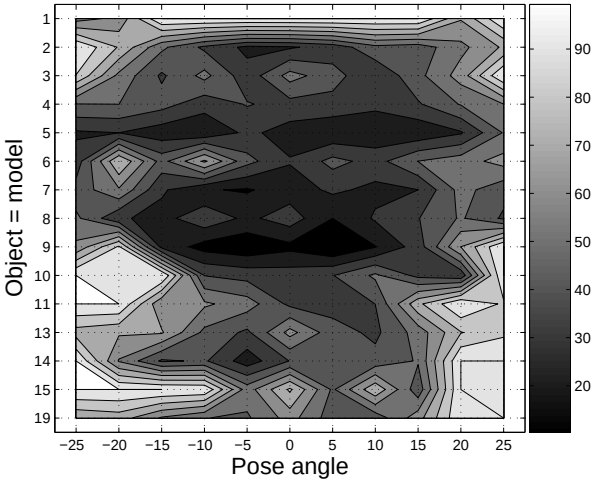


Figure 7.32: BP pose variation plot for the COIL-20 database, using LCV.

the frontal poses (the BP error ranges from 10 to 60), well within the ground truth thresholds There is some falloff for the extreme angles, similar to the lower CC we have seen previously and which probably will affect the acceptance scores too. Objects 5, 7, 8 and 9 have the lowest scores with 5 being the one with the consistently lowest error score for all poses. Even so, between angles of $\pm 15^0$ all objects produce a very low geometric error with the familiar slight drop for an angle of 0^0.

The last plot for the LCV model on this database is the average acceptance graph shown in Fig. 7.33. This is displayed as a 3-D surface for the 15 modelled objects and covers training angles between $\pm 20^0$. We see a near flat surface at over 80% acceptance score for the majority of the objects with a few, isolated basins at 70%. There are three spots where the acceptance falls to a low of 10% for objects 9, 14 and 19 at angles of $\pm 20^0$ which coincides with our observations from the CC and BP graphs previously. Only object 1 has a significant drop in acceptance score for the frontal angles $\pm 5^0, .., \pm 15^0$. This is quite

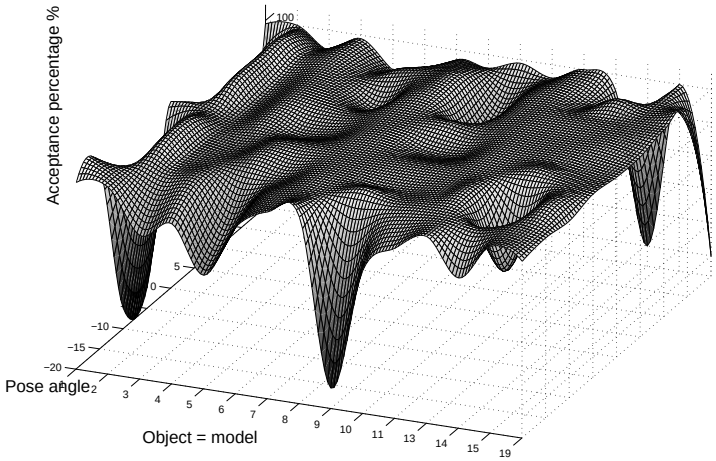


Figure 7.33: Acceptance performance surface plot for COIL-20 database, using LCV.

unusual and cannot be explained by looking at the accuracy of the CC and BP error results. Therefore, we are lead to the assumption that it is perhaps due to erroneously chosen, very high empirical thresholds.

Finally, we show the results from the tests using the AAMs on the same data. First is the RMSE plot in Fig. 7.34 where it is obvious that there is a much higher error than when the LCV was used. This is something we have seen several times in our experiments so far. We therefore expect a lower efficiency due to the (now usual) disagreement of the AAM test results and the ground truth. If we look closer at Fig. 7.34 we see that object 13 still maintains a (relatively) good and consistent performance across most pose angles. In addition, objects 6, 8 and 10 seem to fare slightly better than 2, 5 and 7 in terms of discrepancies between test score and g.t. values. One interesting observation is that the high-RMS spikes for a pose of 0^0 visible in Fig. 7.30 have now reversed into lower-RMS dips. Furthermore, the graph contains many such spikes and dips, and there is no more a smooth transition of the RMS error between pose angles. This might occur because of the "binary" nature of the optimisation algorithm associated with the AAMs. It will either converge at the correct solution or get completely lost, but nothing in between.

We proceed to the CC and BP plots (Fig. 7.35 and 7.36 respectively) where we see slightly lower CC scores than obtained in the LCV case but conversely better BP error values. Objects 7 and 8 have improved scores for most poses in both graphs whereas objects 14, 15 and 19 have now worse accuracy results, something that did not occur in the LCV tests. In general for the AAMs the performance seems to deteriorate more dramatically as we move away from the near-frontal poses. There are a few spikes of lower accuracy in both graphs in particular for objects 5 and 7. Such isolated spikes are most probably associated with the inability of the optimisation algorithm to converge rather than that of the AAM model to capture the pose variation of the object or the error measures to provide a unique, well-defined and low global minimum at these pose angles

We conclude with the acceptance percentage efficiency scores which may be seen in the surface plot

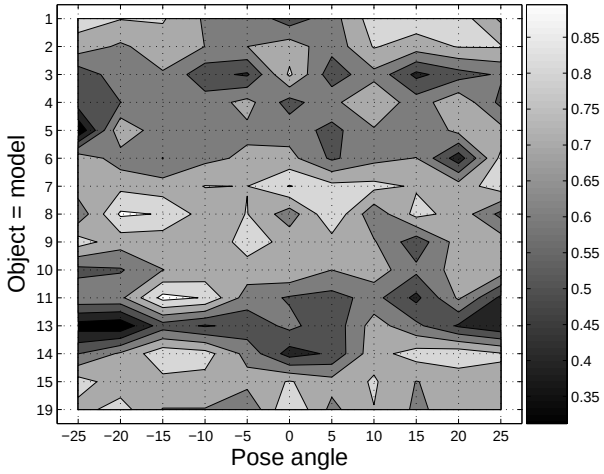


Figure 7.34: RMSE pose variation plot for the COIL-20 database, using AAMs.

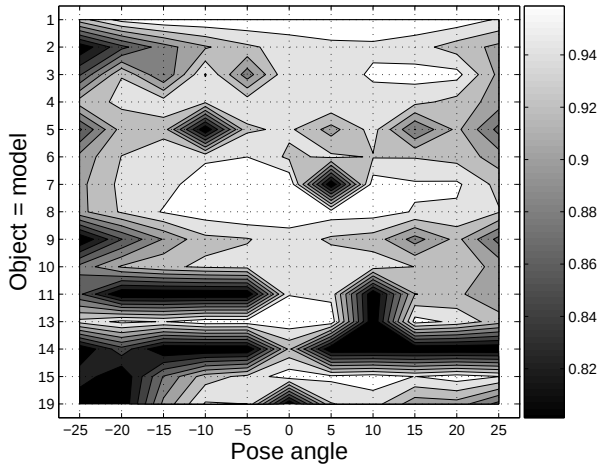


Figure 7.35: CC pose variation plot for the COIL-20 database, using AAMs.

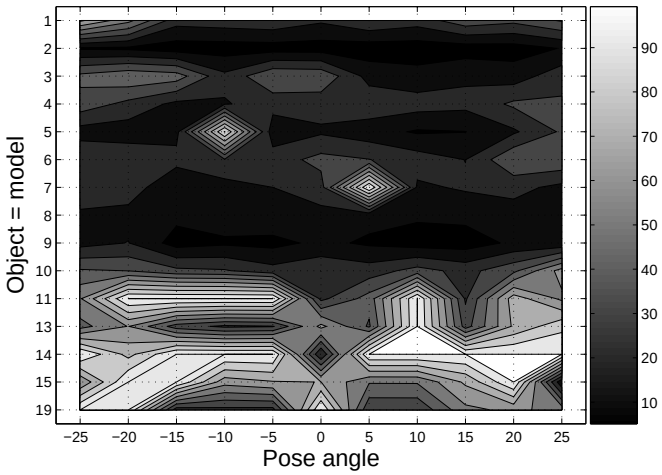


Figure 7.36: BP pose variation plot for the COIL-20 database, using AAMs.

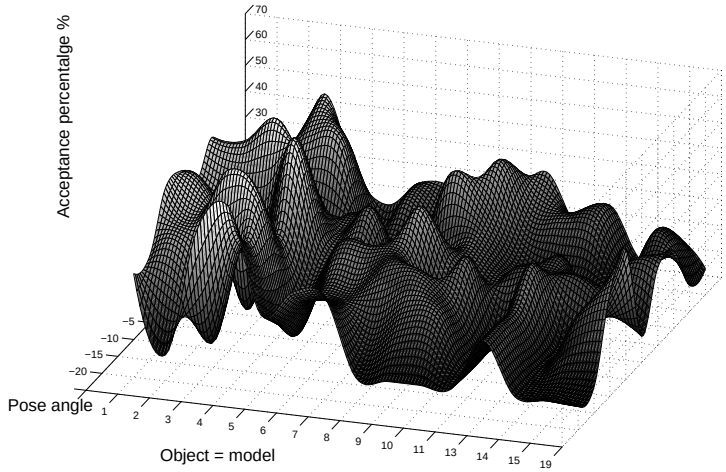


Figure 7.37: Acceptance performance surface plot for COIL-20 database, using AAMs.

in Fig. 7.37. What is immediately obvious from this is that there is a dramatic difference from the LCV graph in Fig. 7.33 and, in particular that the efficiency scores have dropped to 0-40%. This is a similar pattern to what we have seen before when considering the performance of the AAMs which becomes more emphasised at angles representing non-frontal views. Objects 1-4 seem to lead to marginally better performance than obtained from other objects and 14-19 have worse overall scores. The rest, 5-13 are somewhere in between the two extremes.

Based on what we have seen from these tests, we may say with some confidence that the LCV model has comparable intensity and geometric accuracy to the AAM when object identification is concerned. When it comes to pose detection and, especially for frontal and near-frontal viewing angles, the LCV gives a somewhat better CC response than the AAMs while the opposite is true for the BP error. The LCV performs well at the extreme viewing angles of $\pm 20^0, 25^0$, something that the AAM is unable to do, possibly owing to a limitation in the model that would otherwise allow it to deal with untrained pose variation where extrapolation may be required. The efficiency results once again have shown the LCV model to be superior mainly, it seems, thanks to its powerful hybrid optimisation algorithm which produces good acceptance scores even in this more demanding dataset. When AAMs' were used, the overall efficiency has remained low with an additional 10-20% reduction from the levels we have seen in the previous dataset (Table 7.2).

7.4.3 Database 3

The final dataset is the Yale B database which is the most challenging set. This is because it contains real images taken under varying illumination conditions (ambient and spot-light) and also in this case the background model is not available. Instead, an approximation to it is provided by an image taken with the spot-lights turned off and only the ambient light illuminating the scene. This is not a perfect solution since the outline of the object (a person) is still visible and the image portion behind it is obscured but using it nevertheless helps ensure a properly defined objective function is available when the model is

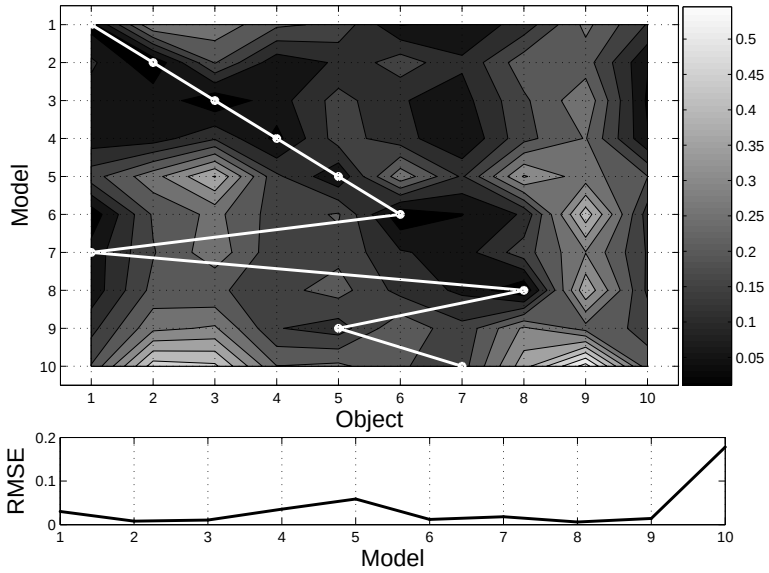


Figure 7.38: RMSE model×object array for the frontal pose, using LCV.

placed over the background.

We start with the identification tests in which every model is compared against each of the 10 objects in turn at the frontal pose (P00) to see if we can find the correct model instance (and its viewing configuration) amongst many different individuals. 100 test runs were carried out in each case and the results were captured in various 10×10, model×object arrays. As usual the first measure to be considered is the RMS error plot (Fig. 7.38) from which we see that there is a well-defined diagonal corresponding to correct identification of model with each object, except for objects 7, 9 and 10. If we compared this to the error-plot in Fig. 7.21 we see that the errors in the two graphs have approximately the same range, something which is further affirmed by the sub-plots from each row. Both sub-plots range from 0.007 to 0.05 except for object 10 in the Yale B database where it is an obvious outlier with a RMSE of 0.17. As we have already seen in Fig. 7.38, models 7 and 9 have also been wrongly identified but their RMS errors recovered in the sub-plot are within the nominal range of the rest of the correctly identified objects and are not obvious outliers as object 10 is.

Also unlike what we found with the COIL-20 database there are no objects here that provide a good match to all the models except perhaps object 7 that matches well with several models. This is a little surprising given that the database contains only faces, but may perhaps be attributed to the more distinguishing differences between the appearance (shape, texture and illumination) of the faces in the Yale B database than we saw for the objects imaged under constant lighting conditions in the COIL-20 dataset. The opposite holds for object 9 which seems to produce a poor matching result for all the models, including its own, and is depicted as a vertical column of lighter intensity.

Next is the CC plot (Fig. 7.39) which illustrates the recognition accuracy of each model when an appearance measure is used. We observe that it has a very similar appearance to the previous RMSE plot with generally a good response (between 0.9→0.98) and a well-defined diagonal for cor-

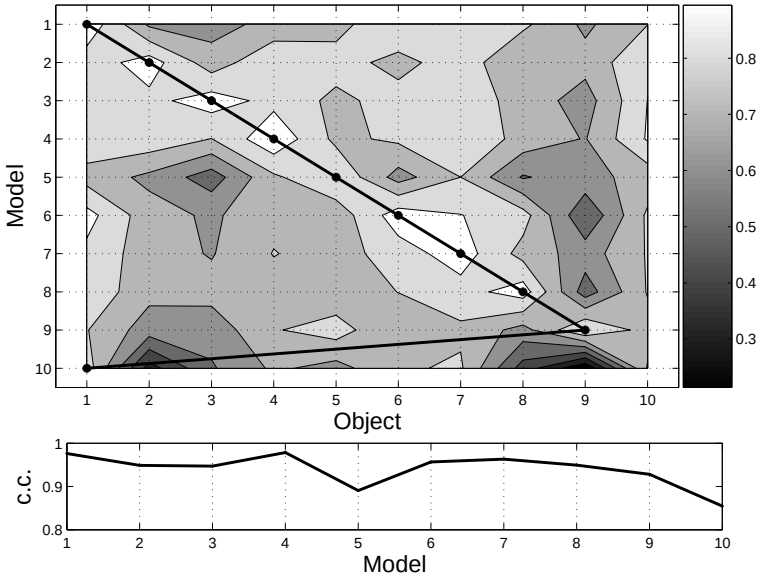


Figure 7.39: CC model×object array for the frontal pose, using LCV.

rect model=object associations, except for object 10 which gives a score of 0.85. Such a poor cross-correlation coefficient leads us to suspect that perhaps this particular object has failed throughout all our tests. However, further work is necessary before we can draw concrete conclusions. Once again, object 9 is the object with the lowest response for all attempted matches where the model≠object.

We move on to the BP plot in Fig. 7.40 which has been capped to a maximum of 1000 to preserve detail. Here we see a very good response and a complete main diagonal for model=object without any miss-identifications. However we note that the BP score for object 10 is quite high (≈ 325) and although this object has been correctly identified, the image synthesis might nevertheless be poor and one that we would regard as invalid. The rest of the models give a response of around 50→100 when they match to their respective objects which, based on our empirical results, are well inside the required acceptance thresholds. As we can see there are no significant false positives in the background region of the graph (model≠object) with just a few isolated spikes in the BP error ranging from 400→600.

The performance graph (Fig. 7.41) reveals a high acceptance ratio of 80-100% when model=object and without any miss-identifications. Only objects 5 and 10 have low performance scores of ≈50% and, if we look back at the CC and BP graphs we can see that those two objects have the poorest overall response even though the matching to object 5 has on occasion been over the convergence thresholdIn general the accuracy and efficiency results we have seen for the identification tests on the Yale B database using the LCV approach are very encouraging except in the case of object 10 which, as already noted, needs to be investigated further.

We can now proceed to the AAM portion of the identification tests and consider the RMSE plot in Fig. 7.42. It is immediately obvious that there are fewer discrepancies in the main diagonal compared to the LCV case (only models 3 and 6 are wrongly identified) but at the same time the RMS error is higher in this case than it was when the LCV was used. This is best seen in the sub-plot where the maximum

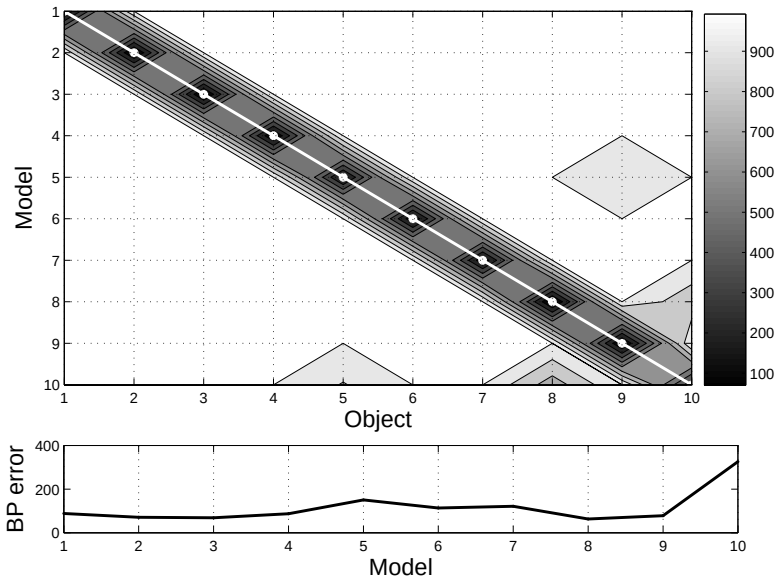


Figure 7.40: BP model×object array for the frontal pose, using LCV.

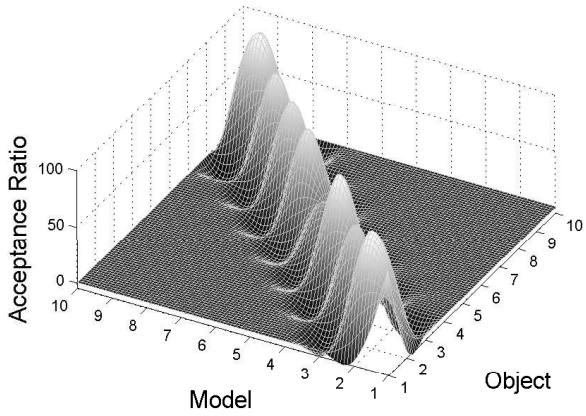


Figure 7.41: Acceptance performance surface plot for Yale B database, using LCV.

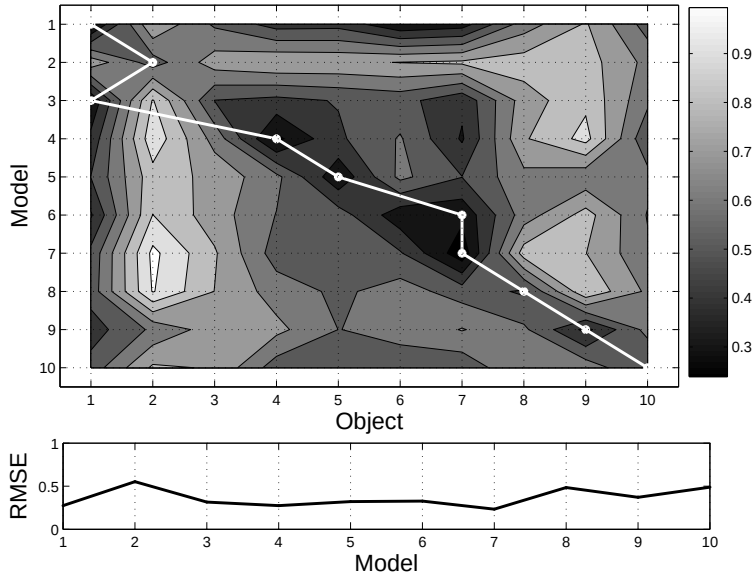


Figure 7.42: RMSE model×object array for the frontal pose, using AAMs.

centres range from 0.2→0.6 whereas in the LCV case they vary from 0.07→0.17. Other remarks we can make from this figure are that object 1 seems to have a good RMS response for most models whereas objects 2 and 9 do not match well with most models in the list. We have already seen this behaviour for object 9 in the previous tests with the LCV.

The next figure is the CC plot (Fig. 7.43). We can see that we have a perfect main diagonal of correct model-object identifications with good, consistent CC scores mostly around 0.9→0.95. Nevertheless the CC scores for objects 5 and 10 still remain low at around 0.85 and 0.8 respectively despite the fact that they have been correctly identified. In addition, the low CC values obtained with the LCV when model≠object for objects 2 and 9 now seem to be somewhat "diluted" and more consistent with the background of the plot in comparison to the very distinct responses that we obtained in the LCV CC plot.

The BP error plot (Fig.7.44) shows a much improved, lower error along the main diagonal including that for objects 5 and 10 and, surprisingly, these two objects have now the lowest BP scores Furthermore the BP errors at the maximum centres range from 25 to 40 which are better than when the LCV was used and also under the empirically derived thresholds. In fact, in our tests so far we have seen a number of different times that the AAMs have slightly better geometrical accuracies than those obtained with the LCV model. The LCV on the other hand gives a marginally higher CC appearance score. We would like to point out that such differences are very small numerically and they do not produce any practical or observable difference in the image synthesis. Nevertheless, this may be of interest from a purely theoretical point of view.

Finally we come to the performance plot (Fig. 7.45) where we see that the acceptance performance score obtained with the AAMs suffers once again with a considerable drop to 10→50% along the main diagonal and with many objects (2, 4, 5, 8 and 10) giving a score of near 0%. The score is 0% elsewhere

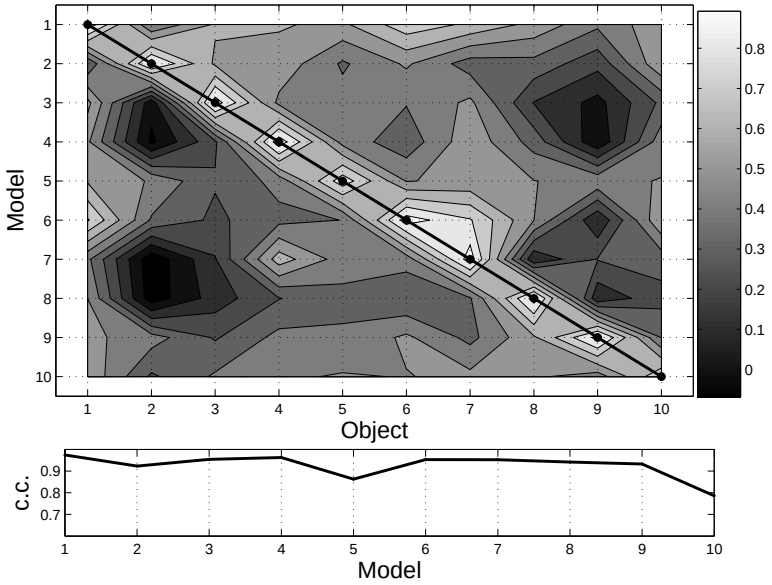


Figure 7.43: CC model×object array for the frontal pose, using AAMs.

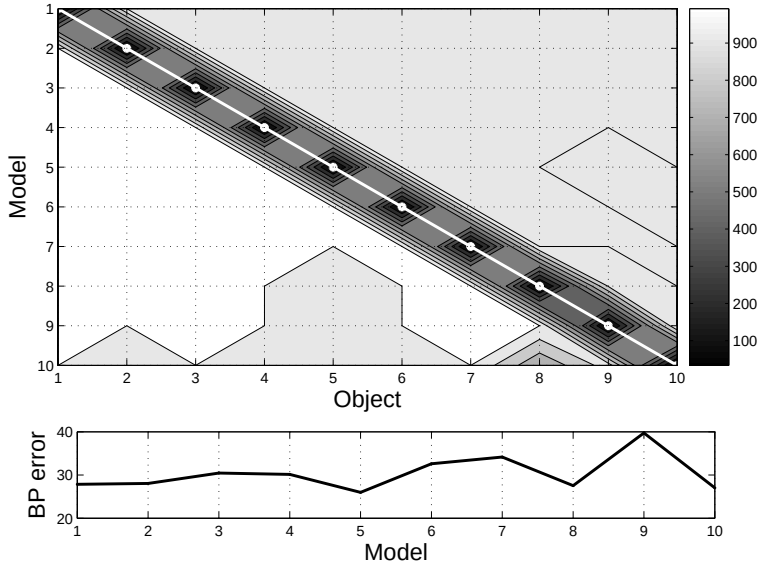


Figure 7.44: BP model×object array for the frontal pose, using AAMs.

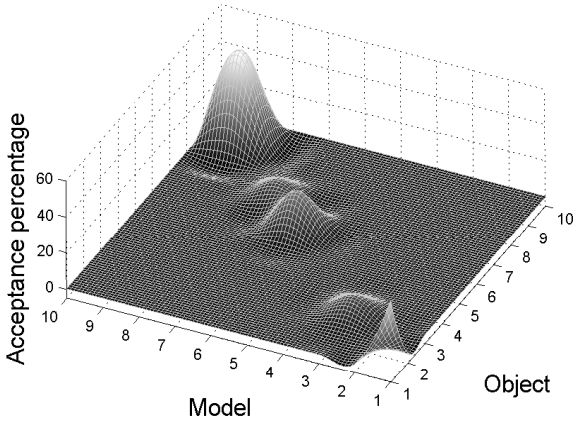


Figure 7.45: Acceptance performance surface plot for Yale B database, using AAMs.

in the plot when model≠object as there are no false identifications (false-positives). The decline is manifested therefore only in the number of times the correct object has been identified rather than in the (true positive)/(false positive) ratio. This is a typical result arising from a poor optimisation algorithm and is not due to some problem with the model itself since the latter would give rise to spikes outside the main diagonal of Fig. 7.45. We also note how the performance of the AAMs seem to decline as the datasets become more complex while at the same time the performance of the LCV approach remains relatively stable.

The rest of the results in this section involve the pose recognition tests, with each model compared against a scene image which contains just one instance of one of the objects that has been modelled. The pose angles here are described in terms of spherical coordinates and are different than in the synthetic and COIL-20 datasets. They also combine pose variations involving rotations about both vertical and horizontal axes (see Fig. 7.8). The pose labelled 0 is the frontal pose along the camera axis. Poses 1, 2, 3, 4, and 5 are approximately 12^0 from the axis and poses 6, 7, and 8 were about 24^0 from the axis. With 9 poses in total (P00→P08) we generated various 10×9 arrays of model=object×pose with each cell being the average value of the chosen measure over 100 test runs. All the objects were tested for all the poses except object 5 for poses P04 and P05 in which where a considerable portion of the face was missing. The model thus could not be accurately built for these two poses selected as basis views owing to landmark points that were missing. Although as we will show in the next few chapters our model (once built) can still be successfully applied to occluded objects, one cannot easily build an LCV model from images in which an object is partially obscured. We therefore decided not to test use of these two poses as basis views for this particular object and to leave the cells blank in the arrays that follow.

Also, if we recall the problems described previously object 10 in the database we can explain now that this was because pose P00 has additional data (i.e. the neck) which is not present in any of the images taken in the remaining 8 poses. This is the opposite problem to that encountered with object 5 and in this case we do not have any missing landmarks. Therefore we can go ahead and build the model for all poses while considering the target images obtained in pose P00 to have additional data that cannot

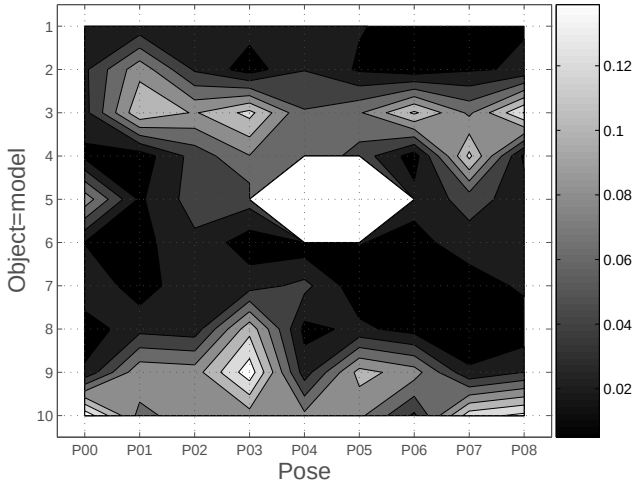


Figure 7.46: RMSE object=model×pose array, using LCV.

be modelled. This would explain the poor CC scores we have seen already which were caused by the additional data whereas the BP error remained good since it only considers the landmark positions and ignores any background data. For this reason we expect a lower CC score for pose P00 for object 10.

We begin with the RMSE plot in Fig. 7.46. Note the missing portion for object 5. The scores are mainly low ranging from 0.01 to 0.1. It is obvious that objects 3, 9 and 10 have the highest error across all the pose angles while the performance for objects 1, 6 and 7 seems to be better irrespective of pose. The results in this graph do not indicate any specific pose angles that consistently produce a very high or low error score but we may remark that for some cases poses P05→P08 seem to give the lowest RMS error. We also note the high error for object 10, pose P00.

The next graph is the CC plot in Fig. 7.47. Most objects give a good response with CC over 0.9 with again no particular pose angles standing out. Objects 3, 9 and 10 are the usual under-performers with the latter having the lowest overall score especially for poses P00 and P05→P08. Object 5 generally does well except for poses P00 and P01. What we have seen so far is a mixed picture with most models recovering the correct pose to within a reasonable accuracy. However, certain models still fail in a number of different poses much more frequently than we have seen for the two previous datasets. This is partly due to the fact that we are dealing with a complex dataset but, we suspect, more importantly because the full background model is not given which is causing difficulty in the optimisation search. We expect this loss of accuracy therefore also to affect the efficiency scores especially for objects 5 and 10.

We proceed to the BP error plot (Fig. 7.48) where we see a similar picture of mixed results. The BP error ranges from 50→150. The geometrical error for objects 5 and 10 seems from these results and those in Fig. 7.47to be a more promising measure for identification purposes than the combined appearance measure. This is because, as we have already mentioned, any additional object features that were not captured by the model (in the image-based template) do not affect the calculation of the

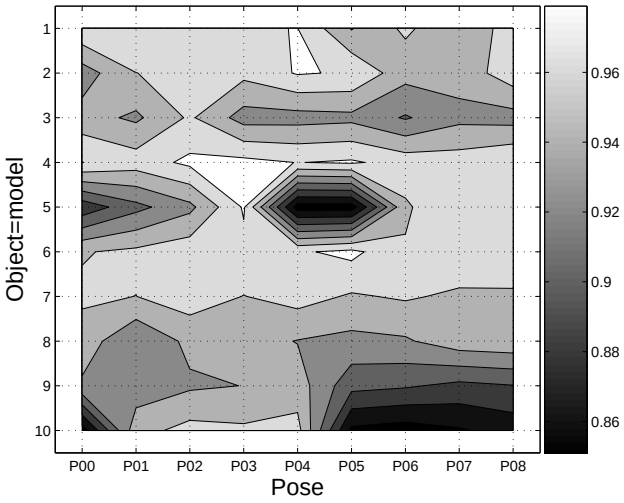


Figure 7.47: CC object=model×pose array, using LCV.

BP error which is only based on the landmark positions. Beyond that remark we do not see anything else that is worth mentioning, except perhaps that objects 5, 6 and 7 have medium BP error responses for poses P00→P01 and that there seems to be a marginally better BP score for poses P02→P04 than P05→P08. It is because of the nature of this dataset with its varying illumination conditions and lack of a proper background image that we have not observed any distinctly high or low responses that span all poses or all objects in the graphs. Observation of such distinctly high or low responses across all poses or viewing angles was a common occurrence when either of the first two databases was used.

The final diagram for the LCV pose recognition tests is the average efficiency graph in Fig. 7.49. Here the acceptance surface is different than the nearly-flat equivalent from the COIL-20 database (Fig. 7.33). We see some low-performance spikes (e.g. with objects 5, 7, 8 and 10) caused by a similar drop in accuracy scores due to the particular characteristics of the Yale B dataset. Despite those few recognition failures at specific poses, overall the acceptance percentage is within acceptable limits ranging from 70→100 for most objects in the set.

Finally, we come to the last stage of our tests on the Yale B dataset which is the question of pose recognition when using the AAMs. First is the RMSE plot (Fig. 7.50) in which we see a moderate RMS error for most objects that is considerably higher than that obtained in the LCV approach. This outcome has been the case so far. Objects 1 and 7 have the lowest scores especially in poses P00, P03→P05. Modelling of objects 2 and 3 remains problematic with a high RMS error indicating that there is a deviation from the ground truth (g.t.) values. We also see that object 10 has a relatively improved good RMS error except for pose P00. Compared with the RMSE plot from the LCV tests (Fig. 7.46) when AAMs are used there are fewer objects that have good RMS values and the difference between objects that produce (relatively) high and low scores has been diminished. We therefore expect to see a definite and considerable drop in efficiency scores for most of the objects in the test set.

Next we look at the combined appearance accuracy in the CC plot (Fig. 7.51). It seems that in

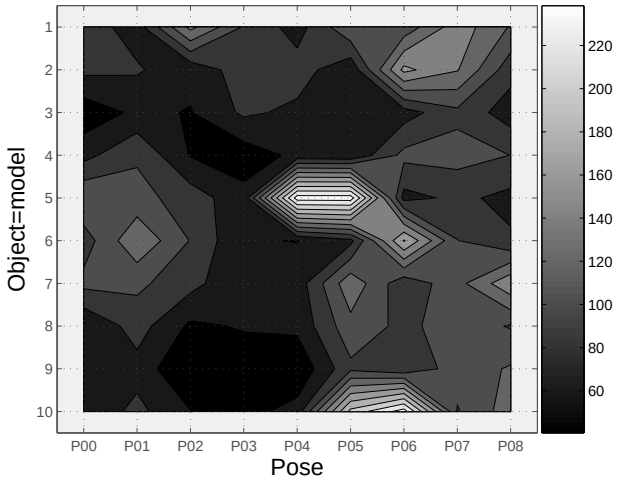


Figure 7.48: BP object=model×pose array, using LCV.

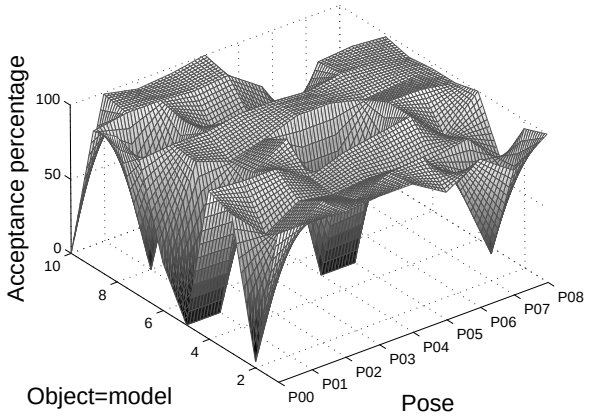


Figure 7.49: Acceptance performance surface plot for Yale B database, using LCV.

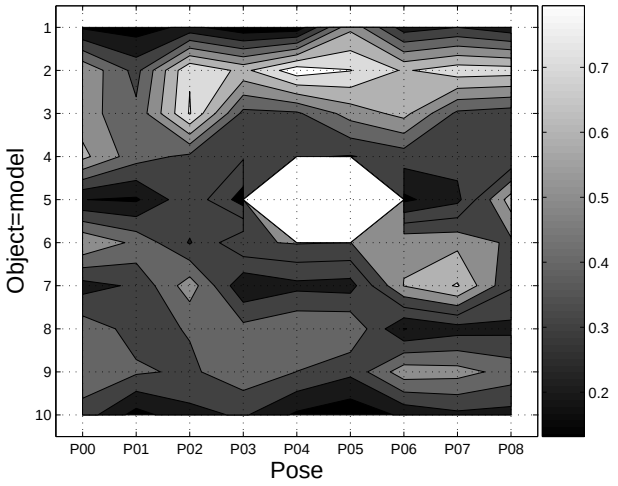


Figure 7.50: RMSE object=model×pose array, using AAMs.

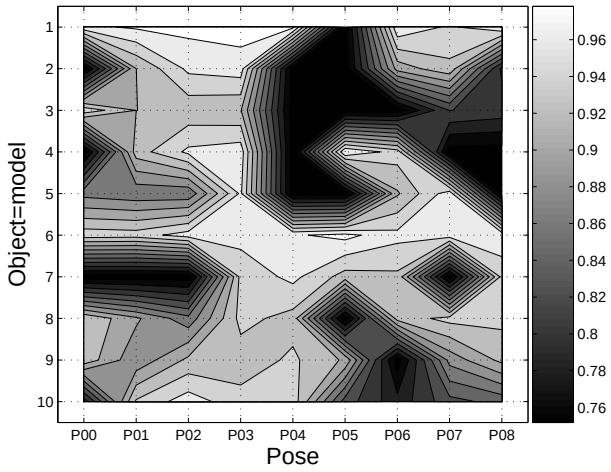


Figure 7.51: CC object=model×pose array, using AAMs.

general and compared to Fig. 7.47 the CC scores have moved into the darker/lower score regions with many objects now having a CC between 0.75→0.8 (e.g. objects 7, 2 and 3) whereas in the LCV case they ranged on average from 0.9 to 0.95. We can also see that results are mixed for most poses and perhaps P03 is the only one that gives a moderately good outcome for all objects with a CC of ≈ 0.9. Also, no object has an invariably low CC value for all poses unlike in the LCV approach where, for example, objects 6, 7 and 8 did.

We move on to the geometrical accuracy scores with the BP plot (Fig. 7.52). At first glance it appears very similar to the previous CC graph with diverse responses. There is no pose with universally good results for all objects, except perhaps once again P03. Compared to the LCV case in Fig. 7.48 we see that for the AAMs object 10 performs much better and objects 4, 5 and 7 do worse for poses P00→P02 but better for P03. In addition, objects 1 and 3 have an improved score for poses P00→P02 but considerably worse scores for the rest of the views. Finally for most objects in the set there seems to be a clearer distinction between the poses P00→P03 that have a low BP error and those with a high error namely P04→P08. In the LCV case this partition is less apparent since there are fewer objects that give such high BP errors.

We end this section with discussion of the efficiency results in Fig. 7.53. When considered in comparison to the LCV results in (Fig. 7.49) it becomes apparent that when AAMs are used many more objects have failed to recover the correct pose with a low associated acceptance ratio of 0-5%. A few of the objects seem to do slightly better such as 1 and 6 ranging from 30-60% for poses P00→P05 and P02→P08 respectively. Overall poses P02 and P03 are the ones with the highest scores at approximately 30-40% unlike the LCV case where all poses gave very good performance results except for a few lower spikes.

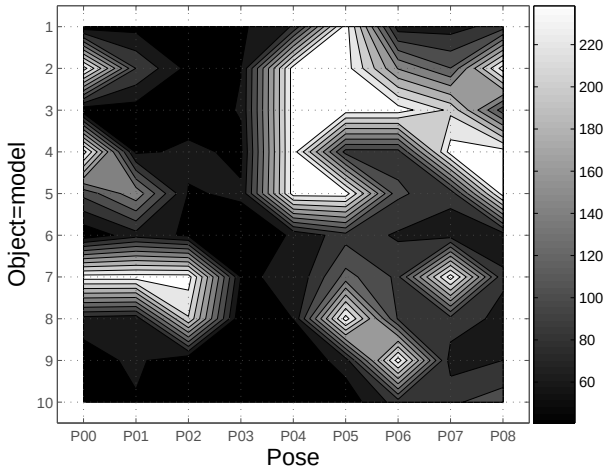


Figure 7.52: BP object=model×pose array, using AAMs.

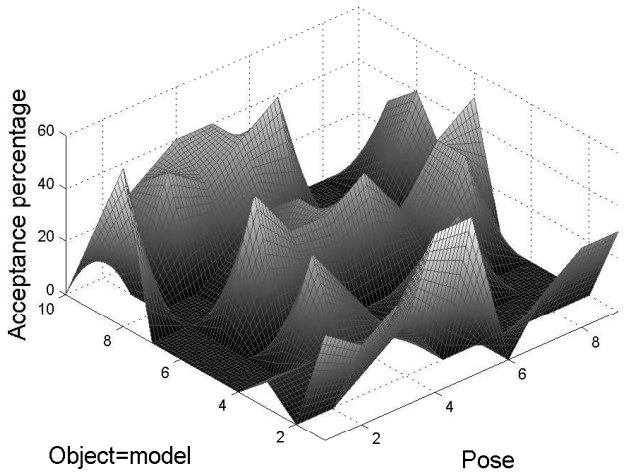


Figure 7.53: Acceptance performance surface plot for Yale B database, using AAMs.

7.4.4 Noise

The synthetic database we have used previously for the evaluation of pose-invariant object recognition is composed of error-free data and so it represents a rather ideal but unrealistic scenario. A more pragmatic approach would be to add a certain amount of random, Gaussian noise that, for example, has not been modelled in the basis views and repeat all the previous experiments in order to assess the extent to which the optimisation results are affected by the existence of noise.

We therefore considered two possibilities: first addition of a moderate amount of noise ($\sigma = 0.05$, see Fig. 7.54(a)) and secondly addition of a large amount of noise ($\sigma = 0.1$, see Fig. 7.54(b)) to the target image pixel values, for both foreground and background pixels. The basis views and pose angle samples are identical to the ones used in section 7.4.1. In addition, we have used similar graphical plots for ease of direct comparison with the noise-free case.

Moderate noise

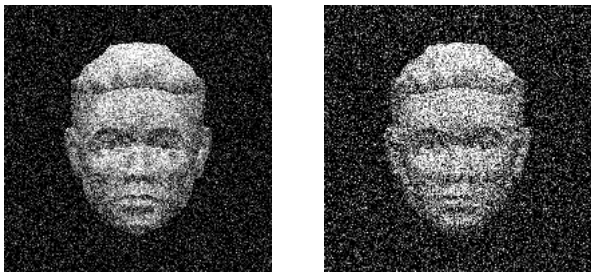
First we examine the RMSE vs MAE graph (Fig. 7.55). As expected in this case the two error measures are higher than in the noise-free experiments implying on average some deviation from the ground truth solutions purely due to the effects of noise. It is interesting to note that the error is bigger for angles $\pm 15, 20, 25$ than it is for smaller angles - something that we did not encounter in the noise-free case considered previously. Also we see that despite the fact that the lowest errors occur at near-frontal poses these are also the angles where there is the largest discrepancy between the RMSE and MAE measures. This means that there is a much higher variation between residuals of the 100 test runs at these angles than for other viewing angles in our test set and that we may subsequently expect to find high accuracy but moderate efficiency scores for these small angles.

Next we examine the average CC graph (Fig. 7.56) for the mode of the sample. Once again it is clearly demonstrated how the addition of pixel noise affects the CC score and yields a significant difference between the observed and empirically derived ground truth values. What is of particular interest however is that all the observations are above the empirical threshold plot[3]. Although the difference between observation and empirical threshold is much lower in this occasion, it is still a very encouraging result, which shows that the optimisation accuracy is not overly affected by the presence of a moderate amount of noise.

Following the above we move on to discussion of the average back-projection error plot (Fig. 7.57). We see that this is very similar to the noise-free plot in Fig. 7.14, except for poses at 0^0 and 5^0 which is where we obtain only quite a low level of accuracy. However, such a close resemblance of the two graphs coupled with Fig. 7.56 itself leads us to the conclusion that these results are geometrically accurate even in the presence of noise.

Finally for the moderate noise scenario we have included two graphs (Fig. 7.58(a) and (b)) that show the overall acceptance percentage for the empirical CC and BP thresholds respectively. At the same time we compare these results with the equivalent acceptance rates from the noise-free case. We note the aforementioned drop in recognition rates of between 5% to 20% at different pose angles for both the

[3]Note that these empirically derived thresholds are different from the ones in the noise-free case and are always related to the current experiments and observations and hence the lower CC values.



(a) $\sigma = 0.05$ (b) $\sigma = 0.1$

Figure 7.54: Synthetic database samples with different amount of random noise.

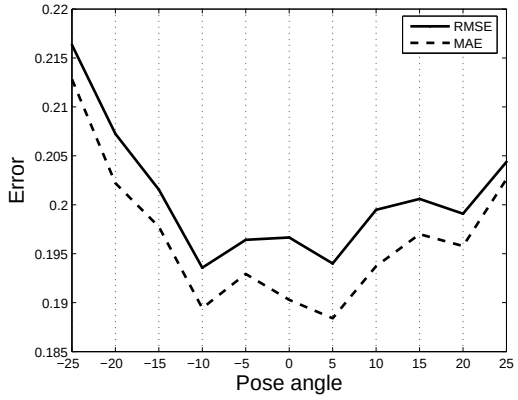


Figure 7.55: RMSE and MAE plots for moderate noise case.

types of threshold used. Of particular interest is the fact that for the frontal poses, the noisy case seems slightly to outperform the noise-free examples. However we believe that this may be attributed purely to the probabilistic nature of the optimisation algorithm and not to some underlying special characteristic of the data.

Before we proceed to the examples with a large amount of noise it should be noted that we have carried out similar experiments with the AAMs on this moderately noisy dataset in order to compare how the active appearance model can cope with such effects. We found that both the RMSE and MAE (Fig. 7.59) have increased considerably compared to the LCV approach (Fig. 7.55) and there seems to be a large discrepancy between the two measures for the AAMs indicating large variations in the error residuals and consequently a drop in the efficiency rate (i.e. reduced acceptance percentage). The latter is most probably due to the inability of the local optimisation algorithm to successfully traverse a noisy error surface. As far as the accuracy is concerned, we see from (Fig. 7.60 and 7.61) that it remains at very good levels relative to those obtained with the LCV approach. As in Fig. 7.18 and 7.19 the graphs obtainmd when AAMs are used exhibit the familiar deterioration at the large viewing angles of $\pm 25^0$ and higher accuracy, in particular of the geometrical error, for the frontal angles.

We end this sub-section with a graphical comparison between the overall efficiency scores of the LCV and AAM for the two empirical thresholds (CC and BP scores) in Fig. 7.62. It is clear that the LCV approach outperforms the AAM approach especially at the angles furthest away from the frontal pose

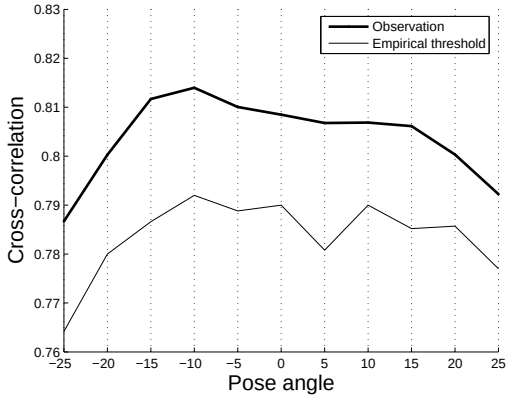


Figure 7.56: Average cross-correlation plot (mode of sample) for moderate noisy case.

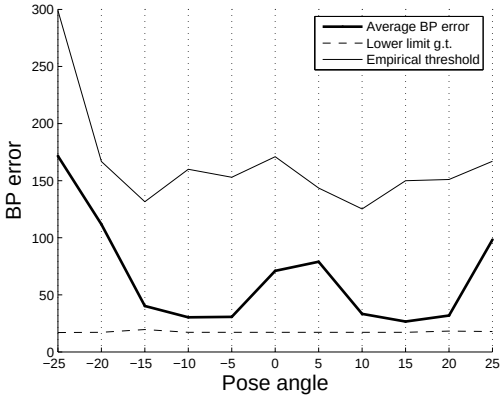
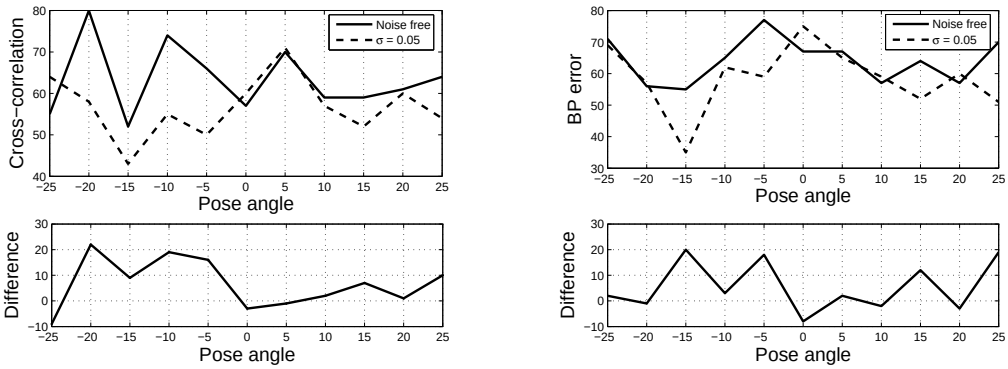


Figure 7.57: Average BP plot (mode of sample) for moderate noisy case.



(a) CC threshold (b) BP threshold

Figure 7.58: Recognition rates comparison using CC and BP score thresholds.

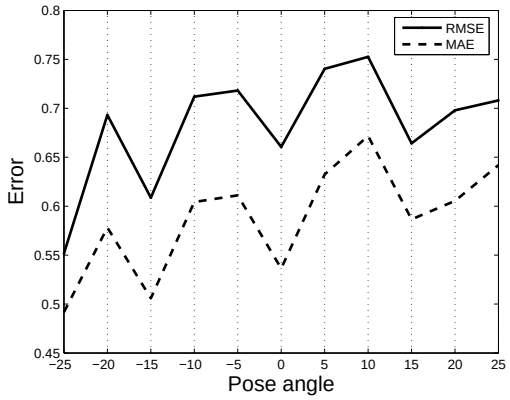


Figure 7.59: RMSE and MAE plots for moderately noisy case using AAMs.

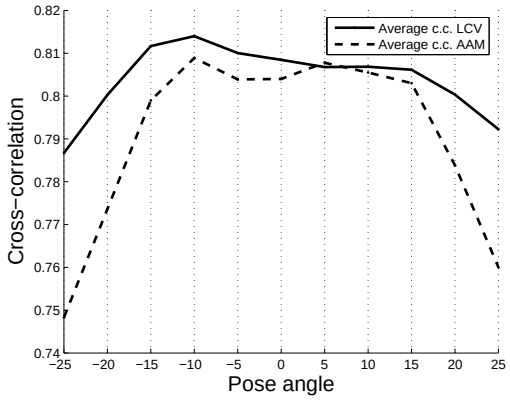


Figure 7.60: Average cross-correlation plot (mode of sample) for moderately noisy case.

for both types of empirically derived thresholds. This is quite the opposite from what we have seen in the noise-free case where AAMs had better recognition scores than the LCV. The accuracy relationship between the two methods seems largely unaffected. We may thus say that the LCV is more robust to noise than the AAM and since only the efficiency differential between the two methods is affected (both the accuracy results degrade by analogous amounts) this robustness is probably due to the superior optimisation solution employed in the former. However, it is necessary to examine the results in the next sub-section when a large amount of noise was added before we draw any additional, general conclusions about the two methods.

Extensive noise

When a large amount of noise was added to the target image as can be seen from Fig. 7.63 the RMSE and MAE errors are higher than they were with moderate amounts of noise and we see that both quantities are almost identical. These observations allow us to make the prediction that there will be an analogous decrease in both the efficiency and accuracy of the optimisation results closely associated with the increased amount of noise. If we had an unexpected, unilateral drop in either the accuracy or the efficiency scores we would expect to see a reduction in the RMSE and MAE errors or an increased disparity be-

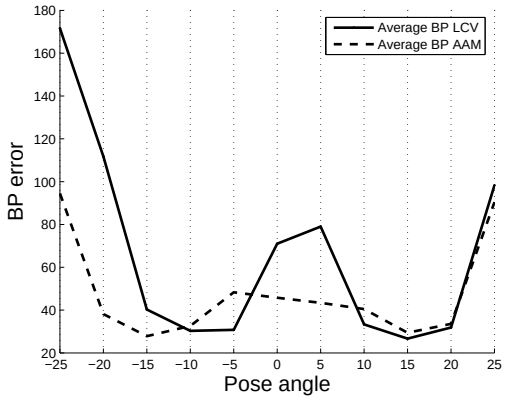
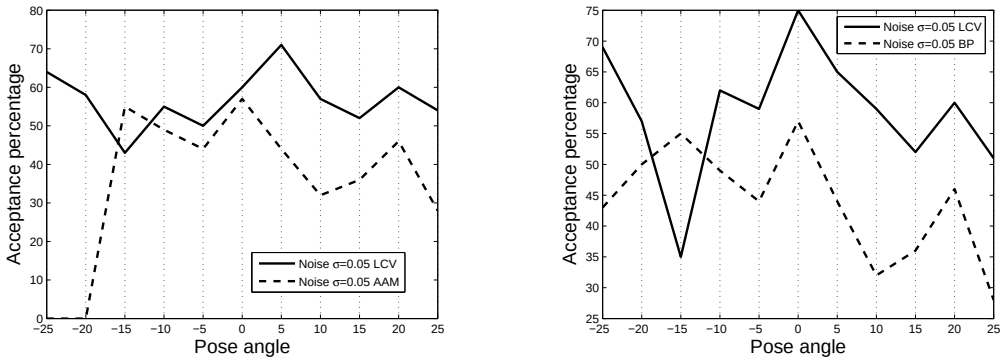


Figure 7.61: Average BP plot (mode of sample) for moderately noisy case.



(a) CC threshold (b) BP threshold

Figure 7.62: Recognition rates comparison between LCV and AAM methods.

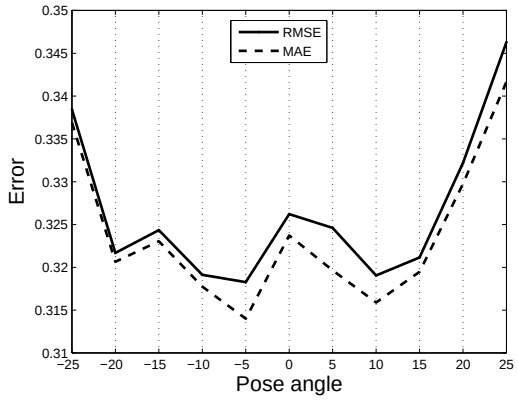


Figure 7.63: RMSE and MAE plots for extensively noisy case.

tween the two but not both. The fact that both of these events have occurred to a limited degree is a good sign that points to a graceful degradation of the object recognition system in the presence of large amounts of noise.

Fig. 7.64 shows the average CC response values and the superimposed empirically derived threshold. In this example we see a lower overall CC score than previously and also that the two plots are much closer together. In fact for some angles (especially pose=0^0) the CC values drop below the threshold for the first time in our tests. Indeed it may be the case that this amount of noise is at the limits of what the LCV model can handle with these optimisation settings. The accuracy scores are somewhat better when the geometrical error is examined (see Fig. 7.65) with all pose angles yielding an average result above the empirical threshold. We should note again that such thresholds are chosen experimentally and on an ad-hoc basis in order to aid the evaluation of optimisation accuracy. They are not definitive or absolute pass or fail rules so we could decide to admit some CC test cases that only narrowly fail provided they have a very good geometrical reconstruction.

If we take account of the above point and use the established empirical thresholds we can generate overall recognition performance comparison graphs (Fig. 7.66(a) and (b)). For both of these graphs we see a considerable drop in the acceptance rates (optimisation algorithm efficiency) in the range of 5-30% for the two types of thresholds. By close comparison to the moderate noise acceptance graphs (Fig. 7.58) we can identify an average 10-20% drop in acceptance percentage as the Gaussian variance increases by 0.05. How well this decrease generalises to other variance values and if indeed there exists a simple, linear relationship between variance and recognition percentage is not apparent and requires more work to resolve. Nevertheless, from the results obtained in the noise-free, moderately noisy and very noisy cases we can identify a gradual and predictable deterioration in optimisation results as the noise in the target image increases. Such a result reinforces our notion that the performance of the LCV although not unaffected by noise is quite robust and has a proportionate decline (or at least not a disproportionate decline) as the amount of noise increases.

We also compare against the AAM search on the same dataset with the large amount of added noise. The RMSE vs MAE plot (Fig. 7.67) shows that even though the individual pose errors have increased

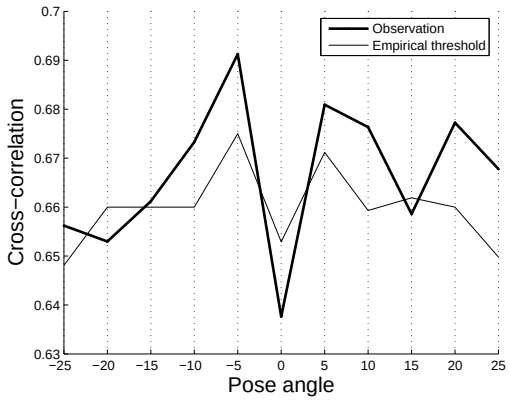


Figure 7.64: Average cross-correlation plot (mode of sample) for extensively noisy case.

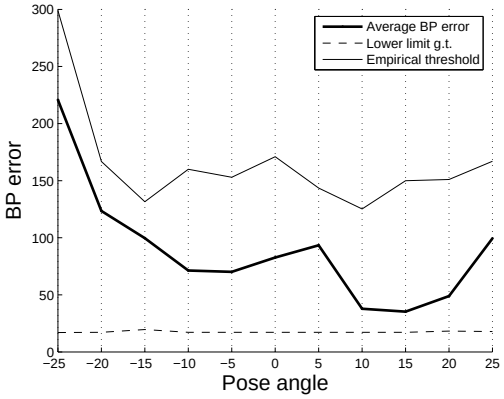
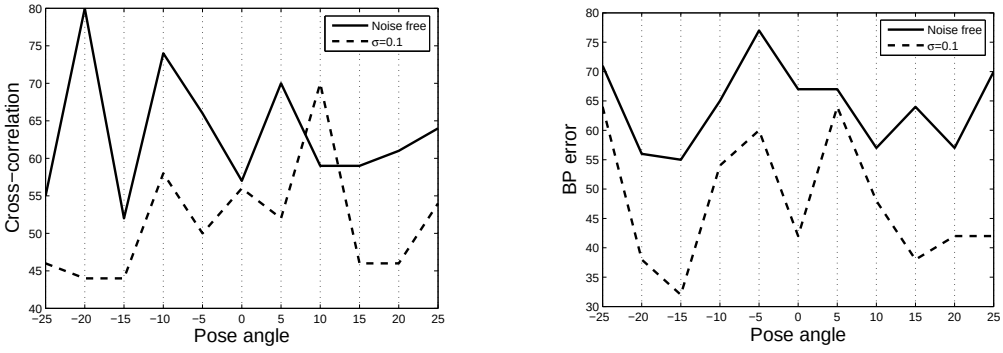


Figure 7.65: Average BP plot (mode of sample) for extensively noisy case.



(a) CC threshold (b) BP threshold

Figure 7.66: Recognition rate comparison using CC and BP score thresholds.

due to the effect of noise on the CC score the relative distance between these curves is lower than in the moderate noise case (Fig. 7.59). In fact we see that there is much less increase in relative RMSE and MAE errors between the moderate and large noise examples using AAMs than there was when the LCV approach was used.

Figures 7.68 and 7.68 compare the average CC and BP responses obtained by use of the LCV and AAM methods respectively. It is obvious that the AAM is as good as the LCV method in the presence of large amounts of noise, except at the large viewing angles of $\pm 25^0$ which have presented a recurring problem for the AAMs. Also when AAMs are used we do not see a drop in the scores near the frontal angles but have a smoother transition between different poses. For the BP errors the results seem to favour the AAMs between $-25^0, ..., 10^0$, however for the remainder angles the two methods have approximately the same level of accuracy.

If we now move on to the average acceptance results for the two methods (Fig. 7.70) we see that the AAM method has a slightly lower efficiency approximately 5-15% less than that of the LCV approach. Note once again the steep drop at the large viewing angles.

In conclusion we may say that the LCV method has an overall good performance in the presence of noise with a predictable degradation when the amount of noise is increased. More specifically, the average accuracy of our approach remains largely unaffected and above the empirically derived thresholds and also quite close to the threshold pertaining to the geometrical ground truth too. In terms of efficiency we see a gradual and graceful drop in recognition rates as the variance of the additive Gaussian noise is increased. Compared to the popular AAM method the LCV performs just as well with comparable accuracy rates especially when the geometrical error of the reconstruction is evaluated. This is because our LCV approach is aided by the powerful hybrid optimisation algorithm (section 6.3.3). We have also seen that the LCV can better model the deformation of the object when limited extrapolation of the viewing angles is required whereas the AAM has some difficulties synthesising the appearance of an object in a pose that has not been seen before and needs to be extrapolated from those comprising the training set.

Our only criticism of the LCV approach is that for the frontal view the model seems to deliver a lower accuracy score and that this is exacerbated by the addition of noise to the target, scene image. As we have said earlier we believe this to occur because the frontal view is where the basis views are combined in equal amounts so any inherent noise in the latter will be enhanced in the synthesised image. If we combine this with the effect of the added Gaussian noise in the target image we get the characteristic, slight drop in cross-correlation for that pose. Conversely, the BP error for the frontal pose seems to be largely unaffected. Furthermore we would like to carry out additional tests in the future to establish more precisely the relationship between increases in noise level and decrease in optimisation efficiency.

7.4.5 Occlusion

This section deals with the effects of un-modelled occlusion on the performance of the LCV approach. These test are designed to represent a replacement occlusion model, whereby an occluding object is placed between the camera and object of interest, and its shape and appearance completely “replaces”

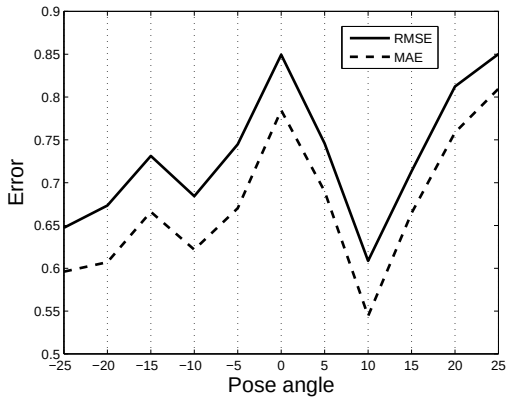


Figure 7.67: RMSE and MAE plots for extensively noisy case, using AAMs.

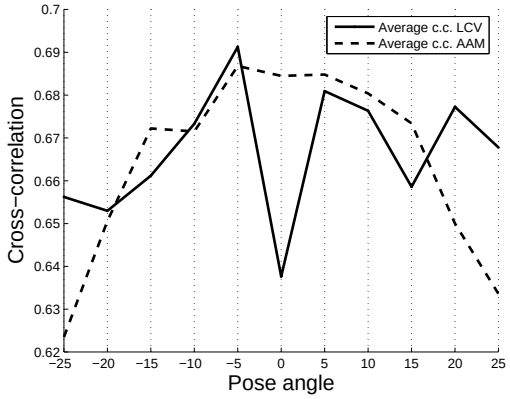


Figure 7.68: Average cross-correlation plot (mode of sample) for extensively noisy case.

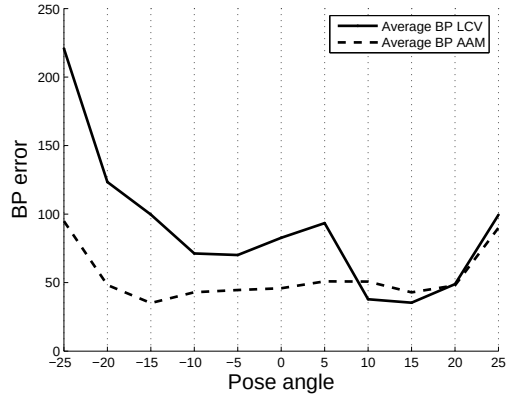
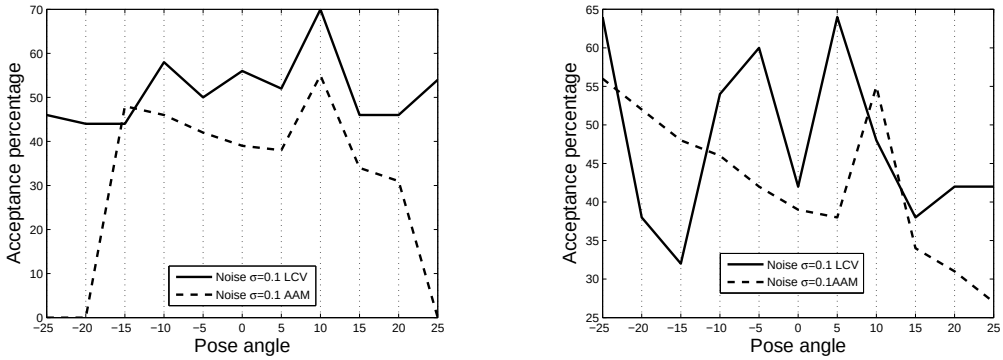


Figure 7.69: Average BP plot (mode of sample) for extensively noisy case.



(a) CC threshold (b) BP threshold

Figure 7.70: Recognition rates comparison using CC and BP error thresholds.

(a) 20% of area (b) 40% of area

Figure 7.71: Synthetic database samples with different amount of random occlusion.

that of the object of interest (over the area of overlap). There are of course alternative models (for example we could have used a semi-opaque or even an object with black, background pixels) that would produce alternative error responses and thus recognition results. Nevertheless, we decided to experiment with the replacement model which is most commonly encountered in real-image scenarios.

Our test data therefore was generated by randomly interposing a white, opaque, circular object within the bounding box of the synthetic head model. We tested two scenarios: first where there was limited occlusion with the size of the occluding object set at 20% of the area of the face (Fig. 7.71(a)); and second with increased occlussion where the circular object was fixed at 40% of the size of the face (Fig. 7.71(b)). The same tests were also carried out with the AAM and compared against our LCV method.

Limited occlusion

As usual we begin with the RMSE vs MAE graph in Fig. 7.72 which is at similar levels to those obtained in the occlusion-free case (Fig. 7.11) although the latter displays a more linear transition between pose angles. There is a large difference between the RMSE and MAE errors in Fig. 7.72 for pose 25^0 but it is too early to identify the reason for this as further analysis is required. It is quite unlikely but still that large number of optimisation tests failed to converge or converged poorly for that particular angle. What is apparent so far is that the 20% occluding object has not drastically affected the performance of the optimisation.

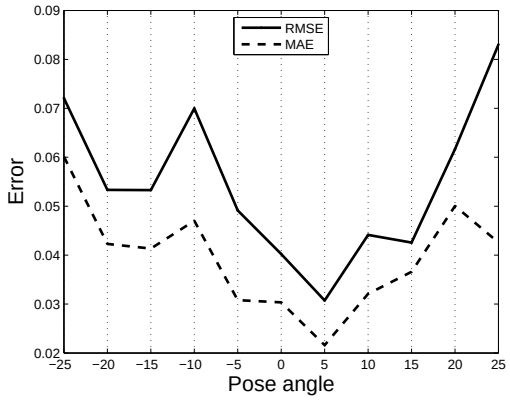


Figure 7.72: RMSE and MAE plots with 20% occlusion.

Figures 7.73 and 7.74 show the average cross-correlation and back-projection plots respectively. By close comparison with the occlusion-free plots (Fig. 7.12 and 7.14) we observe that the occluding object has resulted in an average 0.02 drop in CC scores. Such a minimal drop is to be expected since the image of the superimposed object has quite different pixel intensities than are found in the image of the synthetic head. Nevertheless the average observation score is between the ground truth and the empirically derived thresholds for the majority of pose angles. There is however an overall smaller distance between the observation and empirically derived threshold plots than in Fig. 7.12 indicating a reduction in recognition accuracy caused by the occluding object. Results are much better for the BP error plot since it has an almost identical, if not better, response in the present case than that shown in the plot from the occlusion-free experiment in Fig. 7.14. Since a localised change in pixel intensities by the occluding object does not affect the BP error the latter is a good indicator of the geometric consistency between scene and synthesised views. It is therefore the case that geometric accuracy has remained virtually unchanged in the presence of limited occlusion.

Finally we compare the average acceptance percentage plots using the empirically derived CC (Fig 7.75(a)) and BP (Fig. 7.75(b)) thresholds. We see that except for the two spikes for poses -20^0 and -15^0 where the occluded test scores fall considerably we have a close similarity between the occlusion-free and occluded cases. These spikes indicate low efficiency scores of the optimisation algorithm for these poses. Although this observation is mirrored by a fall in accuracy for pose -15^0, as we have seen in the cross-correlation plot, it does not occur for angle -20^0 or at all in the BP plot. Results are thus rather inconclusive for these poses. However it may be that the empirical error thresholds were erroneously chosen too high for these poses (see empirical threshold plots at these angles in Fig. 7.73 and 7.74).

We now move on to the AAM tests where we see a significant reduction in the optimisation accuracy which is depicted as an increase in both the RMSE and MAE quantities in Fig. 7.76. The fact that both these errors are approximately equal for most angles indicates that the error residuals from the 100 test runs differ significantly from the average ground truth values. The accuracy drop is further pronounced in the average CC and BP plots (Figures 7.77 and 7.78 respectively), where we can see quite a lot of oscillation between good and poor matching scores. This is in fact due to the apparent sensitivity of

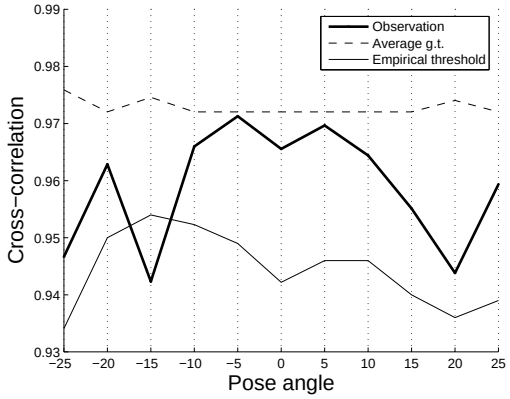


Figure 7.73: Average cross-correlation plot (mode of sample) with 20% occlusion.

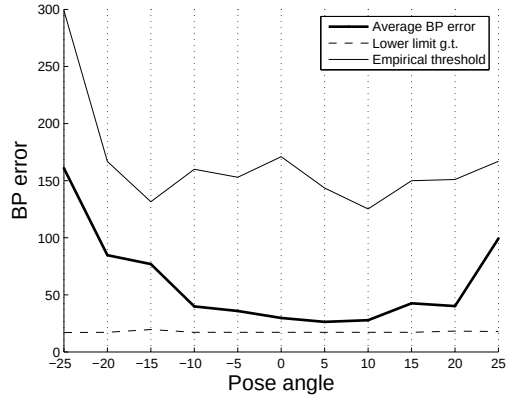
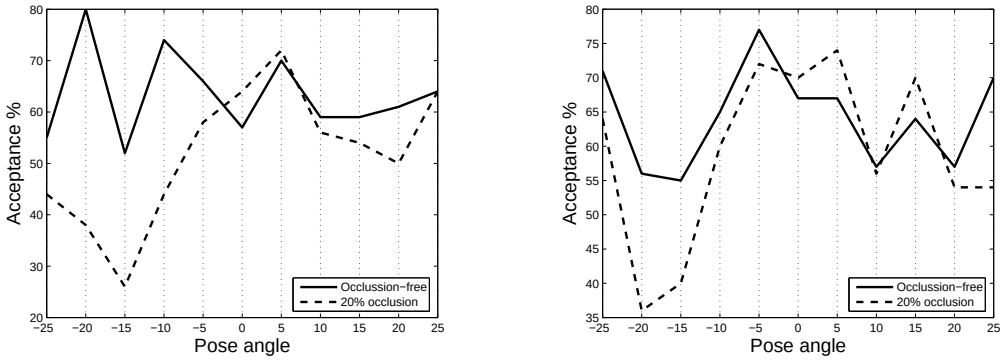


Figure 7.74: Average BP plot (mode of sample) with 20% occlusion.



(a) CC threshold (b) BP threshold

Figure 7.75: Recognition rates comparison using CC and BP score thresholds.

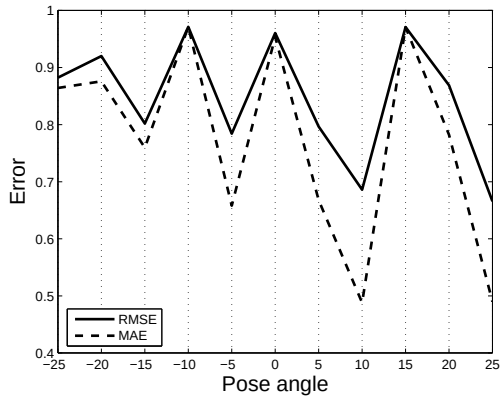


Figure 7.76: RMSE and MAE plots with 20% occlusion using AAMs.

the AAMs to missing data (that may be the result of an occluding object) and not some phenomenon associated with a particular pose angle. In fact, the oscillation at different pose angles is completely random and is determined by the local optimisation algorithm used with the AAMs.

Note additionally, that in this particular case the data histograms are multi-modal with the primary mode caused by trivial solutions[4] (see CC histogram in Fig. 7.79). Thus, even though there is a secondary mode above the empirically derived threshold and even though for some poses the primary mode may indicate a correctly converged solution, when all the tests are examined collectively we can see that the overwhelming response is toward very small cross-correlation values. This explains the appearance of the CC and BP plots.

It would also be reasonable to expect a similarly large decline in acceptance scores. This is indeed the case if we examine the acceptance graphs in Fig. 7.80. The AAM tests have subsided to very low acceptance rates (zero in some cases) for both empirically derived thresholds in comparison to those obtained with the LCV approach for the same, 20% occlusion, dataset. These results are perhaps indicative of the fact that the AAM may not be very robust to even a small degree of occlusion unlike the LCV method. However we should analyse the results from the 40% occlusion dataset before we draw any further conclusions about how the two approaches compare.

Increased occlusion

For the second case where the occluding object is doubled in area we see a small increase in RMSE and MAE errors (Fig. 7.81) in the magnitude of 0.02-0.03 for the majority of pose angles. The two quantities are now much closer together indicating an overall agreement between individual error residuals and the average ground truth. Combined with the low scores it is an indication that the accuracy and efficiency of the algorithm might have decreased slightly but still remains at good levels. There is some increase in the RMSE and MAE values as we move away from the frontal pose but this is normal and has already been observed in the 20% occlusion case (Fig. 7.72).

As far as the average responses are concerned Fig. 7.82(a) and (b) demonstrate that both the CC

[4]The AAM has collapsed to a single point that gives a CC score close to zero when compared to the scene image.

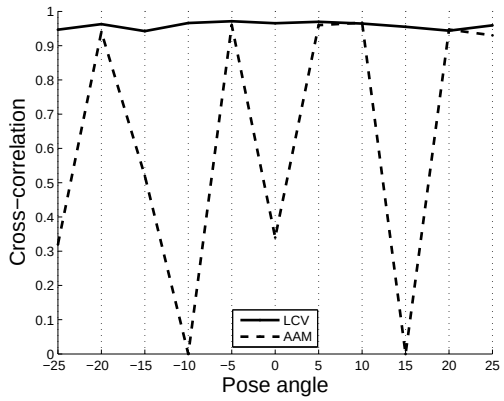


Figure 7.77: Average cross-correlation plot (mode of sample) using AAMs.

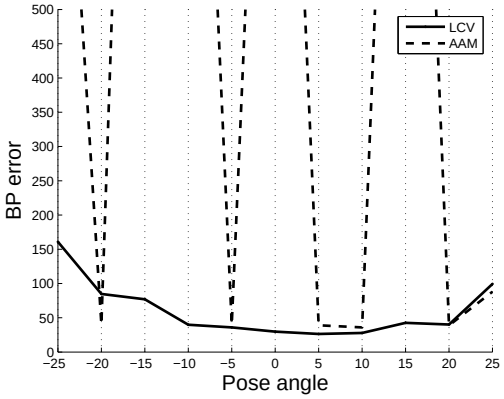


Figure 7.78: Average BP plot (mode of sample) using AAMs.

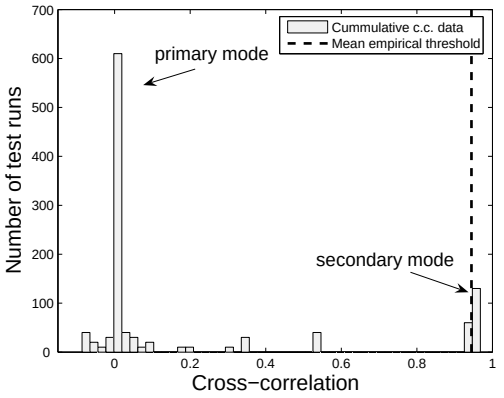
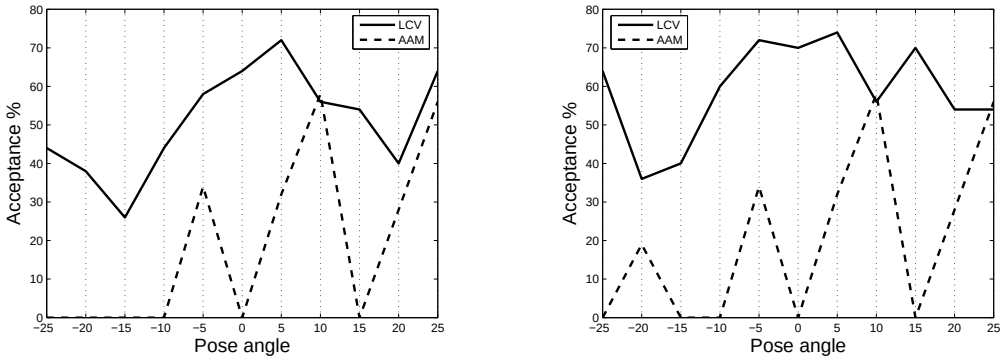


Figure 7.79: Total data histogram for 20% occlusion, using AAMs.



(a) CC threshold (b) BP threshold

Figure 7.80: Recognition rates comparison using CC and BP score thresholds.

and BP scores are adequately above the empirically derived thresholds. The geometrical error seems to be largely unaffected by the increase in size of the occluding object and at similar levels to those obtained when the occluding object's size was 20% of the face. This indicates a good reconstruction of the geometry of the object from the trained model even if a significant portion of the data is missing from the scene image. The CC has dropped from the values obtained in the previous tests but that is a result of the occluding object which directly affects the CC calculation. We see that despite this the average CC scores are obtained from synthesized images that produce visually acceptable reconstructions of the target, scene image although the CC plot is now closer to the empirical threshold plot than it was when a 20% occluding object was used. We may also look at the two histograms (Fig. 7.83(a) and (b)) and see that there are no other significant modes with a good proportion of test runs scoring consistently over the empirical thresholds. As a result we may conclude that the accuracy of the synthesized geometry remains unchanged whereas the combined appearance accuracy has dropped slightly but still demonstrates a robust result given the significant increase in the occluding object's area.

If we now move to the optimisation efficiency determined from the test-run empirical acceptance percentages (Fig. 7.84(a), (b)) we see that they are at similar levels to those obtained when an occluding object 20% of the face size was used. In more detail we see the CC response in this case of increased occlusion is higher than most the responses were in the case of the more limited occlusion for non-frontal poses. For the BP threshold the results obtained with 40% occlusion have a small drop in efficiency which increases for non-frontal viewing angles but is still close to that obtained in the experiments with more limited occlusion. Similarly to the accuracy, the average efficiency rates have not been overly affected by the increased occlusion of the object of interest.

We have carried out the same experiments with the AAM and found that none of the test runs managed to converge to score that would meet the empirically derived thresholds for any of the viewing angles. If we look at the two histograms (Fig.7.85(a) and (b)) we can see an overwhelming concentration at CC$\approx$0 and a shift toward high BP error values. This is a similar to the behaviour in the case of less severe occlusion we have seen before where the AAM cannot cope very well with occlusion of the target object of interest. The drop in accuracy and efficiency scores is highly disproportionate to the increased

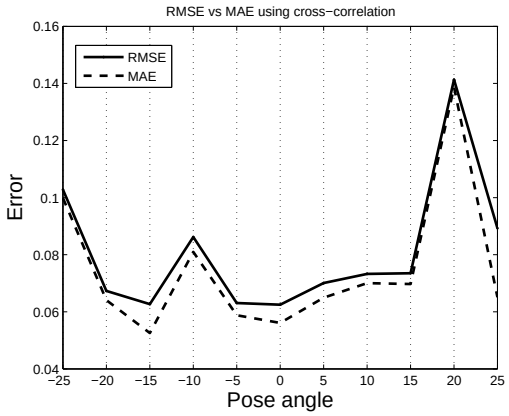
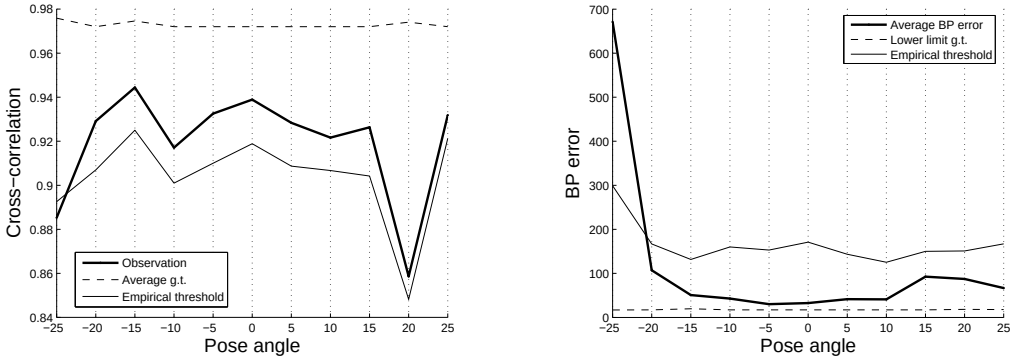
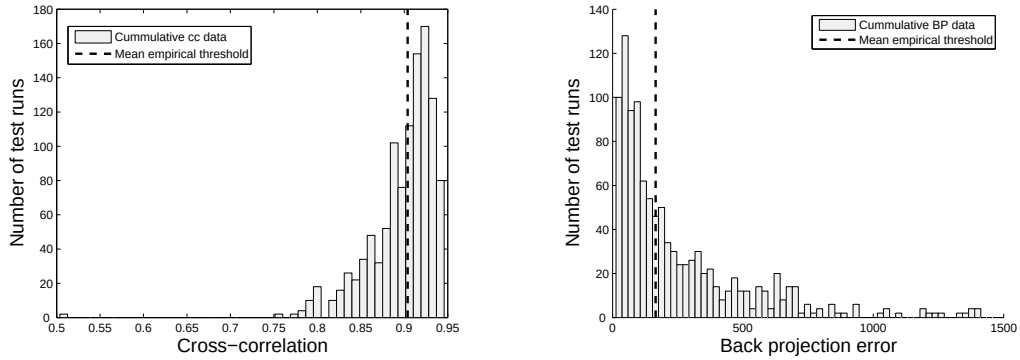


Figure 7.81: RMSE and MAE plots with 40% occlusion.



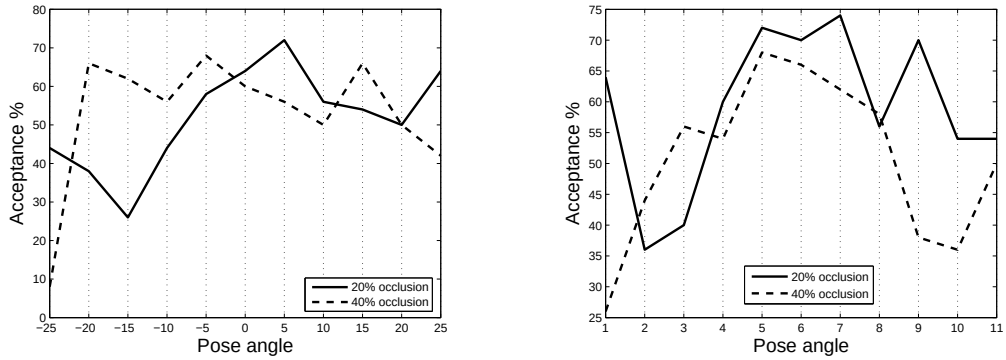
(a) Cross-correlation plot (b) BP plot

Figure 7.82: Average CC and BP plots (mode of sample).



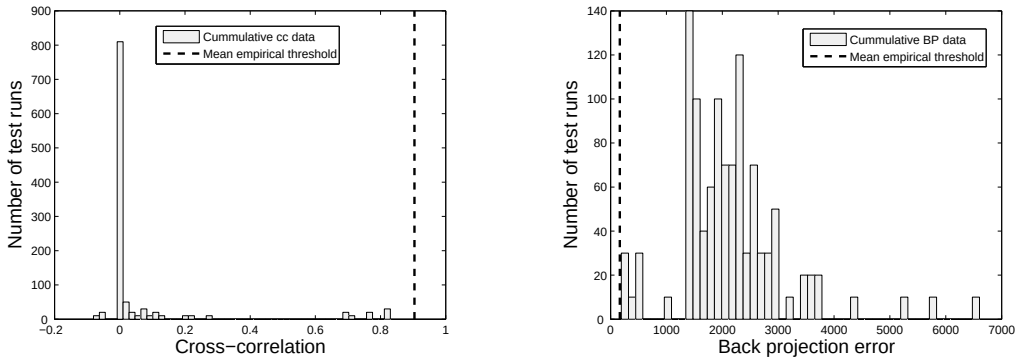
(a) CC histogram (b) BP histogram

Figure 7.83: Average CC and BP plots (mode of sample).



(a) CC threshold (b) BP threshold

Figure 7.84: Recognition rates comparison using CC and BP score thresholds.



(a) CC histogram (b) BP histogram

Figure 7.85: Average CC and BP plots (mode of sample).

area of the occlusion and as a result we may generalise our conclusion by saying that the AAM in the current implementation we have used is not robust to missing data due to occlusion.

On the other hand the LCV approach deals very well with occlusion with very little loss of accuracy and an acceptable minor loss of efficiency. Furthermore, as the amount of occlusion is increased (to 40% of the object of interest) the LCV performance remains largely unchanged. We believe this is because of the fact that the allowed coefficient ranges are learned during training and incorporated into the Bayesian priors which play an important role in the optimisation search process. In addition, the hybrid algorithm, assisted by these priors can avoid trivial solutions and concentrate on areas of the objective function where meaningful solutions are most likely to occur. The AAM when confronted with unknown data (such as that arising from the occluding object) in the vicinity of its search locus is ill-equipped to deal with uncertainty and randomly searches the objective function until it converges to a trivial solution (as we have seen in both the occlusion CC histograms Fig.7.79(a) and 7.85(a)).

7.4.6 Expression

The experiments in this section reflect our attempt to test against the effects of localised, flexible deformations of the object of interest, exemplified in our experiments by changes of facial expression. We considered views of the synthetic head in two different, un-modelled expressions and carried out the

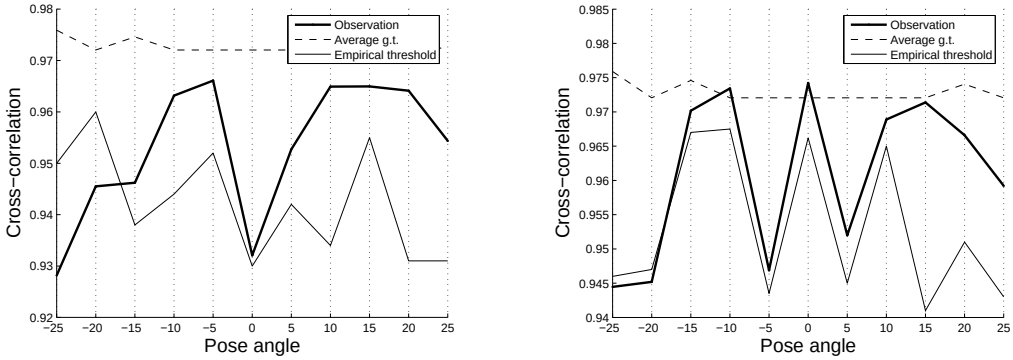
usual kind of experiments using our existing LCV model. Comparisons were carried out against the previous results obtained from use of the LCV approach with target images exhibiting the modelled, natural facial expression. The AAM portion of the tests have been excluded in this case since the AAM cannot model local deformation that has not been included in the training set. We therefore focused on the ability of the LCV method to recover the correct pose alone.

The first point of comparison is between the average CC and BP error plots. We see that due to the change of expression, the CC plots (Fig.7.86(a) and (b)) are lower that those obtained in the neutral (modelled) expression plot (Fig. 7.12) but still above the empirical threshold. Only pose angles -25^0 and -20^0 are marginally below the cut-off point. A similar situation is apparent when we examine the BP plots (Fig. 7.87(a) and (b)) and compare them with the similar graphs obtained for the the neutral expression (Fig. 7.14). The plots for both un-modelled expressions are further away from the g.t. boundary but below the empirical threshold whereas the plot for the neutral expression is closer to the former and thus more accurate geometrically. We also note a slightly better overall score obtained for experiments with the happy expression as opposed to the angry one since the former represents a more localised deformation and a smaller change (e.g. in the lower lip, the chin and the nose) is required to transform from a neutral face to a happy one than from a neutral face to an angry one. This may be a fine point but it is nevertheless clear that the two expressions do not produce exactly the same responses.

For the evaluation of the average efficiency we have included a comparison of the graphs (Fig. 7.88) for the three expressions. As we have already seen efficiency for experiments on target face images in the natural expression ranges between 55 to 80%. It appears linear and stable between different poses without any significant spikes in the curves. The same also applies for the BP threshold acceptance response. For the two un-modelled expressions, it has a larger variation in the efficiency rates ranging from 30% for angles 25^0 and -15^0 to 90% and 100% for poses 15^0 and 0^0 respectively. We see a similar result in the BP plot. We would like to make clear at this point that the empirical thresholds for this particular scenario, were determined based not on how well the LCV model could synthesise the new expression as this would simply be impossible, but on how well it would recover the (visually) correct pose in the presence of localised, un-modelled variation. This should explain why sometimes we observe higher efficiency rates than in the experiments incorporating only pose variation in section 7.4.1. The empirically derived thresholds are thus different but at the same time slightly lower response scores may be obtained because the model cannot match precisely to images of the face with the previously unseen expressions.

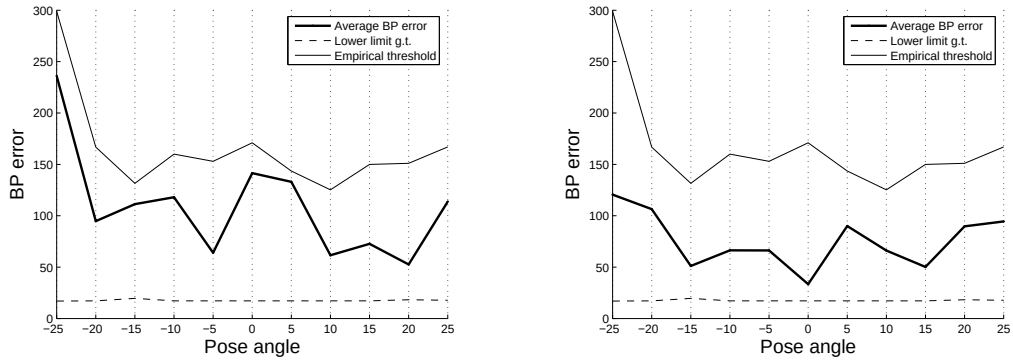
7.4.7 Rotation about a horizontal axis

In this sub-section we present our experimental results on pose variation due to rotation about the horizontal axis. Five samples were used from -10^0 to 10^0 at 5^0 intervals while the pose about the vertical axis remained fixed at 0^0 (frontal pose). With these tests we intend to examine the ability of the LCV model to accommodate a second set of view changes. To do so we need to determine the plausible limits for the 10 coefficients and their optimal combination(s) that would produce valid target image view syntheses. We would also like to compare the results with those previously obtained when the pose of the



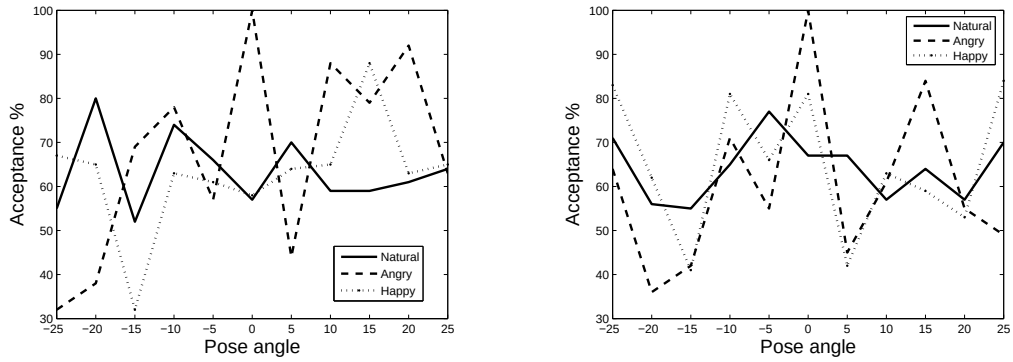
(a) Angry (b) Happy

Figure 7.86: Average CC comparison for unmodelled expressions.



(a) Angry (b) Happy

Figure 7.87: Average BP comparison for unmodelled expressions.



(a) CC threshold (b) BP threshold

Figure 7.88: Acceptance comparison for unmodelled expressions.

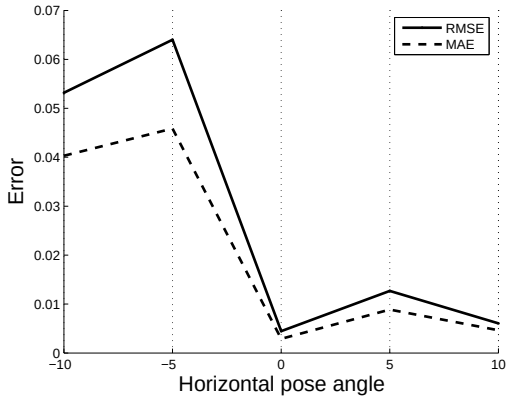
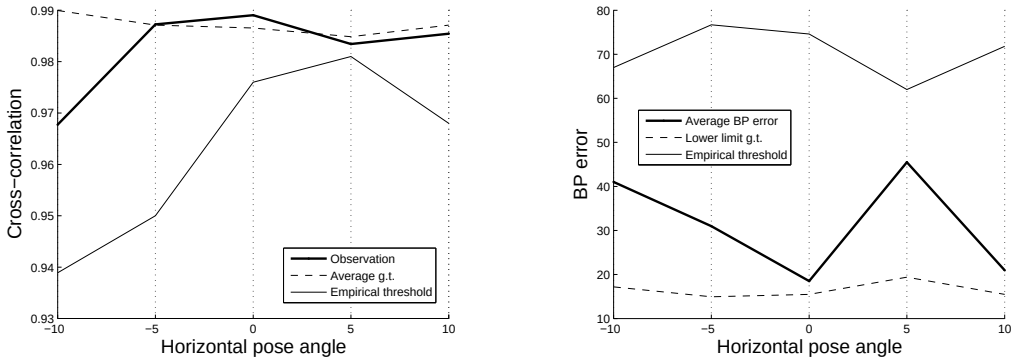


Figure 7.89: RMSE and MAE plots for horizontal rotation.



(a) CC (b) BP

Figure 7.90: Average CC and BP responses for horizontal rotation.

object or equivalently the viewpoint were rotated about a vertical axis and identify any similarities and differences between the two. In the interest of completeness we have also included in this scenario the tests carried out using the AAM.

The first figure we will consider is as usual the RMSE vs MAE plot (Fig. 7.89). We see a very good low error response for both quantities that rises slightly for angles -10^0 and -5^0. If we compare it to the results obtained for rotation about a vertical axis (Fig. 7.11) we see that rotation about a horizontal axis produces much lower error values for the frontal pose possibly indicating that basis views selected along the vertical axis are better suited for synthesis of that particular angle than ones along the horizontal axis. Next come the average response graphs for the cross-correlation score, CC, and for the back-projection errors, BP (Fig. 7.90(a) and (b)). The accuracy results here are very good comfortably meeting the required thresholds for both measures. In particular for the cross-correlation we note that some responses are above the ground truth values. A close comparison to the results obtained for rotation about the vertical axis at the frontal pose (Figures 7.12 and 7.14) reveals considerably higher accuracy scores when the rotation is about the horizontal axis. The average efficiency plot for the CC and BP thresholds is shown in Fig. 7.91. The algorithm produces good results and for some angles close to 90% and 100%. There is only a significant drop for poses 10^0 and -5^0 for the BP threshold.

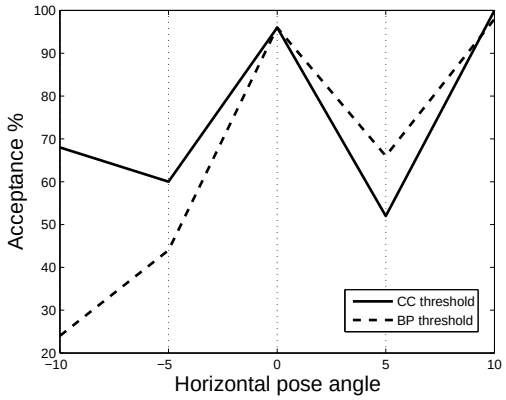


Figure 7.91: Average acceptance comparison for CC and BP score thresholds.

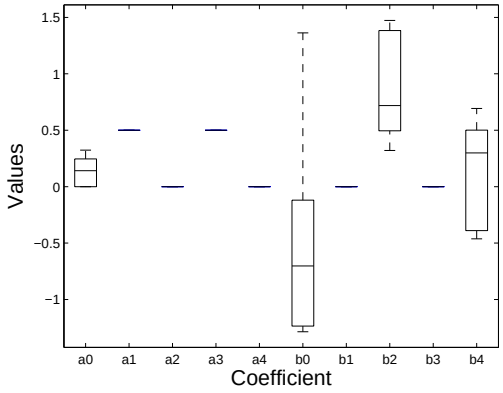


Figure 7.92: Diversity of mean coefficients.

We now take a look at the coefficient diversity plot in Fig. 7.92. Unlike the case for rotation about a vertical axis (Fig. 7.16) here the b_j coefficients are the ones responsible for the pose variation while the a_i are static, except for a_0 which together with b_0 represents translation of the synthesized image. As the translation on the abscissa (x axis) is minimal a_0 is much smaller than b_0. Note once again that these two coefficients have different units (or as physicists say, dimensions) than the rest and so it is not unusual to see a larger variation in therm than in the other coefficients. For the remainder, only b_2 and b_4 vary while b_1 and b_3 are fixed at zero. b_2 and b_4 vary from $\sim$0.3 to $\sim$1.5 and from $\sim$-0.5 to $\sim$0.7 respectively with b_2=b_4=0.5 corresponding to the frontal pose 0^0. The diversity spread for these coefficients is similar to that for a_1 and a_2 in the case studied for rotation about a vertical axis even if the range of angles of rotation was larger in the latter case. It might be the case therefore that the scale of the non-trivial corresponding a_i and b_j in the two experiments are also different. Such differences will reflect the extents to which the appearance of the face changes as it is rotated about the two axes In addition, the translation range captured in b_0 in the current experiments is much smaller than that captured by a_0 in Fig. 7.16.

Finally for the rotation about a horizontal axis we present the results obtained usiung the AAMs. As

usual we begin with the RMSE vs MAE plot in Fig. 7.93. Compared with the equivalent LCV plot (Fig. 7.89) the AAM results show a larger overall disparity between the two quantities while it is noticeable that the former performs better for the poses at $0^0, 5^0, 10^0$. In the average accuracy response (Fig. 7.94(a) and (b)) the AAM has the clear advantage when the back-projection error is measured but it's approximately at the same cross-correlation levels as those obtained with the LCV. It is our hypothesis that when the objection function has an easily traversable error surface, and provided a good initialisation is available the AAM can recover a more geometrically accurate solution than the LCV either because of a more capable model or a better local optimisation algorithm. Note that we have only tested a small range of rotation angles about the horizontal axis so these assumptions might not generalise very well to other situations.

Lastly, we examine the acceptance graph (Fig. 7.95) which should give us a general idea about the efficiency of the AAM in comparison to that of the LCV approach. First we observe that the AAM has the same acceptance results for both the CC and BP thresholds, as we have encountered previously, due to the local optimisation algorithm and the tight optimisation threshold boundaries. We can also see that the AAM model has a good acceptance rate between 70-100% in the same region as that obtained with the LCV model. In addition, results from the AAM do not exhibit the same excessive drop for the BP score at an angle of -10^0. On the other hand the LCV seems to outperform the AAM at the frontal pose on both score.

In conclusion we have built an LCV model for a scenario in which the synthetic face object or viewpoint is rotated about a horizontal axis and have examined its application across the range of the poses $-10^0, ..., 10^0$ with all landmarks being visible at all times. We have seen how the b_j coefficients fluctuate and over what ranges they vary in order to account for the pose variation. This is particularly interesting from a training point of view for the calibration of the Bayesian priors. In terms of accuracy and efficiency the LCV approach seems to perform better for this rotation about a horizontal axis, especially for the recognition of the object in the frontal pose where we can carry out a direct comparison with the similar situation when the pose ort viewpoint is rotated about a vertical axis. We believe that this increase in accuracy and efficiency rates is partially due to the additional descriptive power of the model created from the basis views separated by this rotation about a horizontal axis and the fact that the pose variation is examined over a smaller angular range. In spite of these differences we saw that the diversities of the LCV coefficients are at similar levels to those obtained when the viewpoint rotated about a vertical axis.

The LCV also compares very well with the AAMs in this case with both producing very high CC results although the AAMs are superior when the accuracy of the geometric reconstruction is considered. This may be because the AAM is more capable of capturing the statistical variation of the object's geometry during these particular pose changes or it may be due to the ability of the local optimisation algorithm used in the AAM better to traverse the objective function which may be more convex-like and thereby recover more accurate solutions. In terms of the efficiency both methods produce equally pleasing recognition performance results.

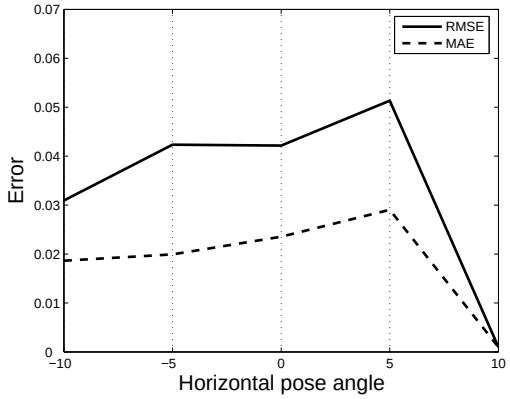
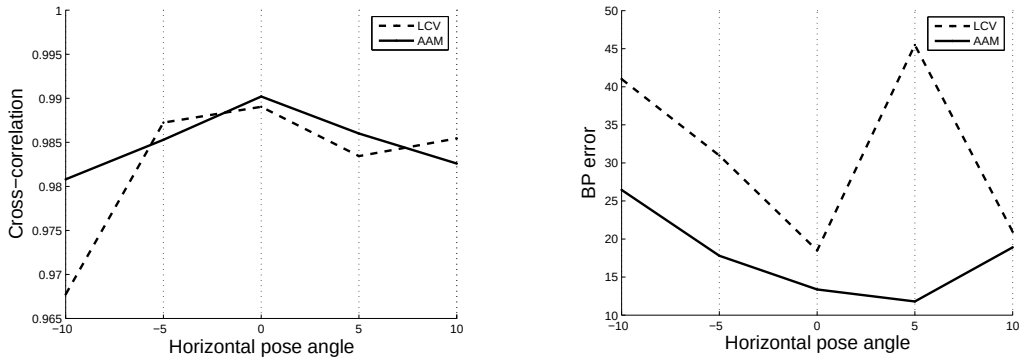


Figure 7.93: RMSE vs MAE plot for horizontal rotation using AAMs.



(a) CC (b) BP

Figure 7.94: Average CC and BP responses for rotation about a vertical axis using AAMs.

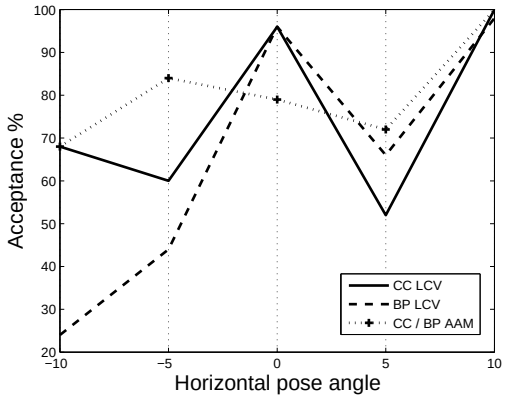


Figure 7.95: Recognition comparison between LCV and AAMs.

7.4.8 Illumination variation

In the beginning of this chapter we have briefly mentioned our intention to carry out a limited number of experiments with the LCV model on images that exhibit non-linear variation in illumination. By non-linear we mean variation in pixel intensities that cannot be fully explained by an affine model which transforms the intensity I to $aI + b$ where a is the gain and b the bias. Such variations may occur due to changes in the location and angle of the light source or sources relative to the camera and object positions - especially if the object of interest has shiny surfaces which can produce specular reflections and can be manifest as cast shadows especially due to non-convex shapes.

We thus would like to evaluate the performance of an LCV system in the presence of un-modelled, non-linear changes in illumination. Note that the evaluation presented is by no means complete in scope or thoroughly examined. However it represents a starting point for study of the effects of illumination variation on our LCV system and may help to identify some of the most general problems or shortcomings of our model that may need to be addressed in future work. We chose to test only the LCV model in this case and not to compare with the use of AAMs since the latter specifically models changes in appearance (which includes implicit illumination changes). If therefore such changes are quite close to what is modelled by the training set and may be accurately interpolated by the AAM we would expect the correct object view to be easily recovered and thus that the AAM would have an advantage over the LCV. The latter does not explicitly model the appearance variation but tries to approximate it from what is known in the basis views.

For our tests we have considered the Yale B database which contains examples of non-linear illumination variation for all the objects in the set. This is achieved with the use of 64 light sources that can fire individually and are set-up in a configuration relative to the camera as shown in Fig. 7.96. The locations of the lights are given in spherical coordinates with azimuth (A) = elevation (E) = 0 being the camera frontal view. We began with an LCV model of the frontal view (P00) for the first object in the database (B01) and with the illumination source at A=E=0. Then for all the 64 scene views of that object in that pose we tried to recover the object configuration and especially the pose angle. The averaged results from 100 test runs for each scene view are give in both flat and surface plot form in Figs. 7.97 and 7.98 for the CC and BP errors respectively. Note that Fig 7.98(b) has been restricted to a maximum BP error of 500 in order to preserve the level of detail at the lower BP values. Also we have used bilinear interpolation to generated values between the samples computed in order to create a smooth surface to better aid visualisation. The centres of the light source locations for which we have exact results are plotted along with the surface data.

Based on these results we can make some interesting observations. First we see that the best response for both measures is not at (0,0) as we might have anticipated but at (7.9, -13.4) and (7.9, 32.4) for the CC and BP plots respectively. This may be explained in part by our misplaced expectation that the maximum cross-correlation should be at (0,0) or in other words at exactly the same location as the scene image for which we trained the model. This is because the LCV does not contain an illumination model but instead tries to approximate the appearance based on the estimated distance of the object from

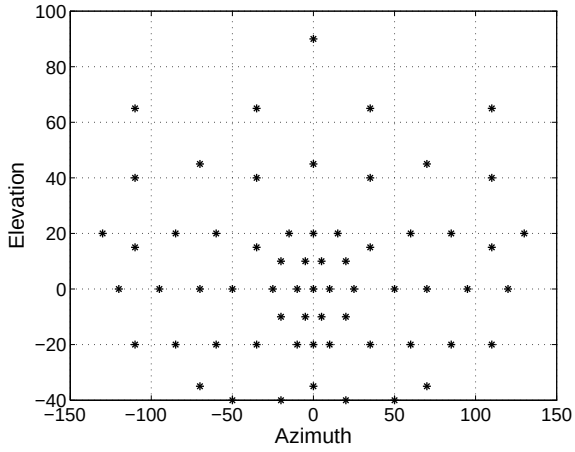
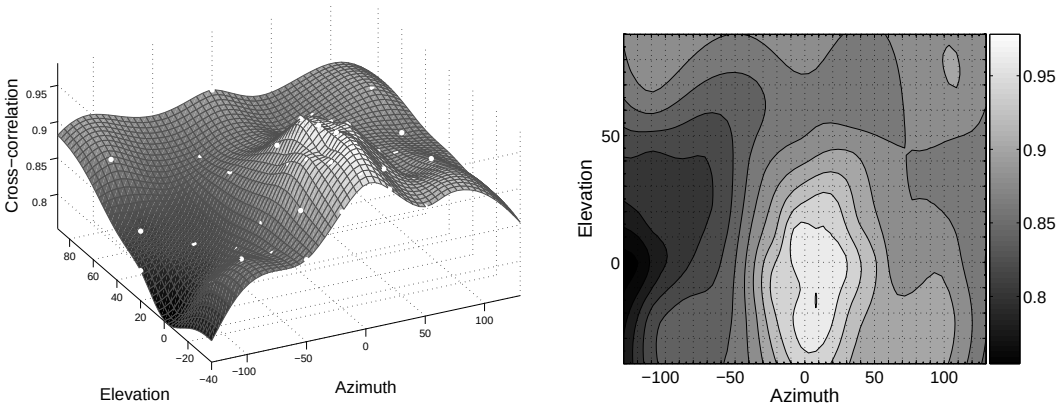
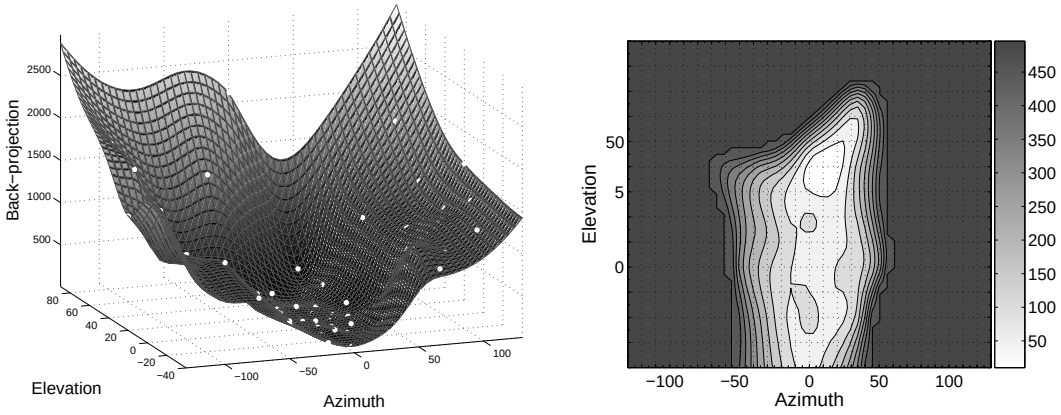


Figure 7.96: Position of illumination sources relative to the camera.



(a) CC surface plot (b) CC contour plot

Figure 7.97: CC response under non-linear illumination variation.



(a) BP surface plot (b) BP contour plot

Figure 7.98: BP error response under non-linear illumination variation.

the basis views. Therefore if specific illumination data is not present in the basis views then it cannot be synthesised in the novel view. The fact that the error is maximum (or minimum) for light sources other than (0,0) simply indicates that the lighting conditions in the two basis views matched more closely to the scene at (7.9, -13.4) than (0,0). This is possibly due to small specularities and cast shadows that may have been present in the basis views because of the disposition of the object of interest, light sources and camera and the fact that the images were taken under spot lights and not only under ambient lighting.

A further observation is that the location of the minimum is not at the same place in the two graphs (Figures 7.97 and 7.98). It is hardly surprising from what we have seen in our results so far that the CC and BP graphs do not always agree. What is more interesting is that the two basins of attraction have different shape, size and location. We notice that the BP is much larger and wider but also has a relatively flat bottom bearing in mind that the ground truth BP thresholds for that pose range are between 50 and 115. This means that quite a large array of lighting configurations will give a low BP error score or phrased another way, the LCV can detect the correct shape in many different illumination settings. These range from approximately $-30 \rightarrow 40$ in the azimuth to $-40 \rightarrow 55$ in the elevation and form the oval shape in Fig. 7.98(b).

For the CC things are a bit more complicated. The basin of attraction looks smaller and much narrower but in this case it is not easy visually to determine the correct convergence. The reason for this is that a localised change in pixel intensities affects the CC match score in a very unobvious and unpredictable way and as a result it is therefore possible that while the synthesised object is geometrically accurate to produce a very low CC value. In addition we cannot chose an empirical cut-off threshold since we are dealing with non-linear changes in appearance and the correct matches cannot be separated from the incorrect matches by a single line. Nevertheless short of checking each result individually we can attempt to select a threshold based on a (traditionally) high CC value and assume that all the examples which meet it represent correct solutions. This approach of course can miss many other valid solutions with lower CC scores. We can see this for example where the BP error plot is at a minimum and at the same position the CC plot ranges between $0.86 \rightarrow 0.88$ values that under constant illumination would signify a very poor synthesis of the target image. It is better therefore to use the BP error plots as an aid to judge accuracy rather than the CC.

One final comment that we would like to make is that both surfaces have a convex-like appearance (more pronounced for the BP error surface) with many good solutions when the light source is close to the camera view axis which gradually get worse as we move away to more extreme lighting conditions both in elevation and azimuth. Moreover there are no significant local minimum spikes anywhere in the surfaces indicating perhaps that although linear changes in the location of the light sources produce non-linear illumination effects in the images they also produce an error surface with simple characteristics which could be modelled and predicted. Nevertheless we decided not to make any specific assumptions about this relationship since we have employed a smooth interpolation technique for the surface visualisation that might make any definite observations and hypotheses somewhat inaccurate.

What is important to take away from these results is that the LCV model can recover the correct pose

not only in the single configuration for which it has been but also when many other similar light sources are located nearby even though they can produce substantially different scenes in term of appearance. This is demonstrated by the wide and flat basins. Also, the error seems to deteriorate in a predictable manner as the light source moves away from the camera view axis and causes heavier cast shadows and localised reflections on the object.

Much more work is obviously required to determine the exact influence of a varying light source on the scene appearance and how this affects the behaviour and performance of an LCV model. It would be ideal if such work could lead to an extension of the LCV approach to include a basic non-linear illumination component.

7.5 Summary

In this chapter, we have carried out a detailed evaluation on the performance of our LCV system in the presence of pose variations, using 3 image datasets of increasing complexity. In addition, we run a large number of pose detection experiments with added noise, occlusion, illumination and expression changes in order to determine how well our system can cope with more realistic situations.

We have shown that our LCV object recognition approach achieves its design objectives of accurately and efficiently recovering the correct pose and complete configuration of the object in a scene with varying characteristics, using both real and synthetic images. Our examination into the accuracy capabilities of the algorithm involved experimentation with different combined appearance and geometry-only measures (RMSE, MAE, cross-correlation and landmark back-projection error) and detailed comparison against ground-truth and empirically chosen thresholds. The tests demonstrated a notable performance with results in close proximity to the thresholds and with little actual accuracy deterioration when progressing to more demanding datasets.

In terms of efficiency performance, defined here as the number of times our tests have terminated within the convergence thresholds and expressed as the percentage of the total, we have seen very promising results in the region of 80-100%, only falling by a small amount for the more difficult Yale B database. We have also established that the LCV approach is quite robust to the presence of a considerable amount of unmodelled noise or occlusion with a small and acceptable drop in efficiency and accuracy rates that increases gracefully and predictably as the amount of noise is amplified or the occluding surface area is enlarged. As far as the changes in expression are concerned, we have seen that although the LCV cannot model these localised variations, it can cope very well and maintain its good performance against changes in appearance.

Furthermore, our system manages very well both in the detection of the correct pose for a fixed object but also as an identification approach for different combinations of models-objects given a fixed pose. In the former we see that the system can approach the correct solution with very few localisation errors, and in the latter with next to none false positive and negative matches.

All the above experiments were re-run using the AAM approach and the results compared with our method. This was done not for determining which of the two systems was better, since our method uses a much more powerful optimisation algorithm, but in order to use the AAMs for the baseline measure

that it is and see how much more effective and accurate our method was in comparison to this tried-and-tested approach, and as a result against other recognition systems that have used the AAMs as a measure in the past. We showed that the two methods have on average comparatively good accuracy results, with the LCV being slightly superior in the combined appearance accuracy (i.e. cross-correlation), while the AAMs performed marginally better when the geometric error was measured. The big difference was in the efficiency rates, where the optimisation algorithm comes into play. Our experiments demonstrated that the hybrid approach gives consistently hight convergence rates and can cope with increasingly complex data, unlike the local minimisation scheme that the AAMs employ, which cannot scale very well when the optimisation problem becomes more demanding.

Certainly this evaluation is by no means complete, and further test are necessary in order to make more robust and generalised conclusions about the efficacy and appropriateness of our approach in both theoretical and real-life, practical applications. Nevertheless, these experiments, carried out in a structured and systematic fashion, tried to cover as much of the test ground ass possible with particular emphasis to 3-D affine, extrinsic pose variations. For the requirements of this thesis (i.e. initial appraisal of the accuracy and efficiency in controlled settings and publicly available data) we believe that we have gone some way into addressing the questions posed in the first chapter. Further experimentation may always be carried out in future work using a larger number of test and data sources in order to fully evaluate a larger domain of different scenarios.

Chapter 8

Conclusions

In this chapter we provide a brief summary of our work together with a review of the main contributions. We proceed with a critical evaluation of our method and argue whether or not it has met our original objectives and if the main hypothesis of this thesis has been addressed. We end this chapter with a discussion on the most important limitations in our work that may be addressed in the future.

8.1 Research summary

We started this work with the intention of examining an approach to the problem of recognition of 3-D objects via a small number of 2-dimensional intensity images while at the same time avoiding the tasks of feature extraction and correspondence during the on-line, model matching stage. In particular, we wished to examine the possibility of using the linear combination of views theory to build a framework and solve this specific problem using realistic, real-image data.

Our first step was to examine the problems associated with the basic feature extraction approach, mainly those of feature extraction and correspondence. For the former we looked at various well-known methods such as edge and corner detectors [Canny (1986); Harris and Stephens (1988)]. For the latter, we discussed techniques such as the interpretation tree [Grimson (1990)] and the RANSAC algorithm [Fischler and Bolles. (1981)] designed to alleviate the computationally intensive correspondence match. It soon became apparent that these are significant problems that cannot be solved to an adequate extent in a practical computational time-frame or without considerable manual input during runtime. Since such object-to-model matching greatly relies on precise feature extraction and establishment of the correspondences (and indeed if these two requirements are met beforehand then matching is a fast, straightforward and relatively accurate process) we decided to avoid any such dependencies and explore a different approach whereby the feature extraction and matching stages have been combined into a single task resembling a template matching approach.

In this way the whole model image is considered a single, multi-dimensional feature that deforms according to some predefined transformation in order to match to the scene view. As a result our search is performed over the transformation space, which is usually much smaller than the original feature or correspondence spaces. In addition, the model building stage has now been further simplified We initially looked at the 2-D object recognition case as a stepping-stone in order to identify and solve some

particular problems in a more manageable set-up before proceeding to the more complicated 3-D case. We considered a 6 d.o.f. affine transformation and used a prototype template containing both grey-level and boundary information. It was later observed that owing to the specific characteristics of the error surface[1] there exist a large number of trivial solutions with very good matching scores and also many local optima when the model is placed over the background. These two problems make search for the correct, global solution very difficult even for the most sophisticated of optimisation algorithms.

Solution to both these problems required the introduction of probabilistic constraints to avoid the trivial solutions when for example, the transformation would cause the template to shrink to zero area, and also to regularise the error surface over the background regions. In order to develop these constraints we separated the affine transform into independent parametric transformations and associated a prior probability with each parameter, thus building a Bayesian inference model. Additionally, we explored different matching metrics including the smooth Huber norm that has a continuous second derivative and can be used with gradient-descent-type optimisation algorithms. Furthermore, it can have a linear response over the background area and thus produce smaller matching error residuals, which are easier for an optimisation algorithm to traverse. Our research then delved into the specifics of the scale transformation as one of the transformations that caused most problems with the occurrence of trivial solutions and, given the assumption that the prior should have some relationship with the distribution of the underlying parameter, we attempted to find the best model for the distribution of image object scale amongst a set of commonly used parametric distributions. In the end, our tests determined that the lognormal distribution produced the best fit and we used this as the scale prior. The full Bayesian model was then tested on various real-image samples and produced very encouraging results even when using a local optimisation algorithm.

Before we moved onto the 3-D case we explored the use of a simplistic, explicit model as a way of illustrating the importance of incorporating the statistical variation of the background area and of regularising the error surface more effectively. We found that incorporating the background is a necessary step if a valid probabilistic interpretation of the matching process is required and also that by doing so one can avoid some of the trivial and spurious solutions. It is however difficult to come up with a perfect model of a complex and cluttered background and unless the target image background is provided the error surface will be rugged with many local optima. This is why we decided to focus more on the regularisation effects of the Bayesian model and the use of a powerful optimisation algorithm to avoid such problems.

Once we were confident with our solution to the 2-dimensional problem, we progressed on to 3-D objects and applied our new knowledge about the specific characteristics of template matching to this new scenario. We began by building a complete recognition system which combines an image synthesis step with an optimisation search-and-match approach. The system synthesises new images using the LCV theory to calculate the correct image object geometry and a piecewise affine interpolation method to cater for the pixel intensities. For the matching we used similar metrics as in the 2-D case such as

[1]These are things which we have encountered many times throughout our research and so consider them to be related to the matching metric and the transformation T used as opposed to a particular data source.

cross-correlation and SSD and we briefly experimented with the use of a mutual information metric.

Just as in the 2-D example we built a Bayesian model for the 3-D case. It was not however possible to suitably decompose the 3-D LCV extended affine matrix into individual, distinct transformations and so we assumed a generic mixture model and assigned a Gaussian distribution to each the 10 LCV coefficients. We then isolated certain transformations (e.g. 3-D rotation of the object of interest or viewpoint since we were interested in pose changes) and recovered the corresponding variation of the coefficients. Based on this information we chose the means and standard deviations of the 10 prior distributions to mimic that variation. Thus for example, coefficients that were almost constant were assigned a very narrow prior with very small standard deviation while others that had larger variation were given an almost uniform prior with a high standard deviation.

Our next research task was to choose an appropriate optimisation algorithm with particular emphasis on the ability to recover a global optimum - usually a minimum - (or at least to get as close to it as possible) without the requirement of a good initialisation or excessive restriction on the parameter boundaries since we had designed the Bayesian priors to take care of any necessary parameter localisation. Efficiency and overall execution speed was a concern but not of paramount importance at least in this proof-of-concept stage that our work represents. We looked at various well-known local and global optimisation algorithms and tested them against synthetic and progressively more complex real-image datasets.

The result was that a hybrid approach, which combined an evolutionary global method (SOMA [Zelinka (2004)]) and a local, deterministic algorithm (the restarting simplex [Zografos and Buxton (n.d.)]) proved to be the best choice to compromise between accuracy and efficiency because it included the localisation performance of the global method and the fast refinement capabilities of the local approach. Based on that outcome we decided to use that optimisation technique in all our subsequent experiments with the LCV object recognition system.

The final part of our study involved the testing and evaluation of the LCV system on real and synthetic datasets. We carried out a large number of structured experiments on three different databases under pose variation but also considering the existence of noise, occlusion and changes in expression (on a face example to represent un-modelled intrinsic variation of an object) and illumination. In addition, we used the Active Appearance Model [Cootes et al. (2001)] method as a general benchmark in order to judge how well our approach was at solving the coupled pose-recognition problem, and in effect how it compares to other relevant strategies that have used AAMs in a similar fashion. The tests have shown that the two methods have on average similarly good accuracy results with the LCV being slightly superior in the combined appearance accuracy (i.e. cross-correlation) while the AAMs performed marginally better when the geometric error was measured. The significant difference was in the efficiency rates, where the hybrid approach gives consistently higher convergence rates and can cope with increasingly complex data, unlike the local minimisation used in the AAMs.

8.2 Critical evaluation - Remarks

This work has examined the challenging task of image-based, multi-view object recognition and managed to address a number of associated problems by employing the LCV theory. With the addition of a regularising Bayesian prior and a powerful optimisation algorithm we managed to build a complete system for recognition of objects that exhibit extrinsic variations, such as pose changes relative to the camera. This is the main achievement of our research; a 'proof of principle' that this approach can work.

More specifically, we have demonstrated that using the LCV system gave us the ability to detect objects of substantially different shape and intensity characteristics in a variety of poses. We have shown our approach to be capable of dealing with datasets of varying complexity both in terms of the foreground object, but also more importantly, the background. Many hundreds of experiments were carried out on publicly available datasets of real and synthetic images, the vast majority of which have highlighted a very good system performance in terms of accuracy and efficiency that degrades gracefully and predictably as the experimental data gradually becomes more complicated.

We also illustrated that our method compares very favourably with the AAM approach which may be regarded as a baseline when aimed at solving approximately the same problem. In more detail, the LCV can recover a similarly accurate solution to a correctly converged AAM which in actual terms is very near the globally optimal solution. On the other hand, owing to the more powerful optimisation algorithm used, the LCV is able to reach the correct solution much more often than the conventional AAM approach we adopted.

We have seen that our recognition system can deal with a considerable amount of un-modelled Gaussian noise present in the scene or target view with the accuracy remaining at high levels and the overall efficiency diminishing in a predictable fashion relative to the noise level. Similar results were observed when we introduced an occluding surface in front of the object of interest covering up to 40% of the object's surface. The system was able to find the correct near-optimal solution the majority of times and with the average efficiency steadily dropping as the occluding object became larger. The accuracy was mostly unaffected relative to the chosen the empirical thresholds. Additionally, we demonstrated that the system is largely robust to localised, non-affine (flexible) changes in the object's shape (e.g. changes in expression in a face example). Even though the LCV system cannot itself model and synthesise these intrinsic shape variations the overall recognition performance has proven not to be strongly influenced by their presence.

Further to the above our tests on the illumination variation examples in the Yale B database have indicated that our system, although it does not explicitly include an illumination model[2] but synthesises a new image based on the information present in the basis views, is flexible enough to correctly recognise the object in a number of similar (but not identical) lighting configurations. In other words, where we might have expected the solution error surface to have a very narrow and deep basin of attraction (the narrowness representing the single or very restricted illumination solution and the depth the considerable difference in magnitude from incorrect solutions), we have seen that it is actually the opposite. There

[2]We have however experimented with a rudimentary affine illumination model for the background and, in addition, note that the cross-correlation coefficient is invariant to affine changes in pixel intensity.

appears to be a wide and shallow basin of attraction with a flat bottom that points to a solution space of adjacent lighting conditions (in terms of light source positions given in spherical coordinates) that our system in its present form is able to recover sufficiently accurately. We believe this to be due to the notable extrapolation abilities of the LCV system which mean that such solutions do exist and they can be found with the optimisation algorithm.

Apart from these main findings we have also identified a number of secondary themes from our research relating to model-based object recognition in general. First is that a full-background model can effectively regularise the error surface when an adaptive template is used and especially when the template model is positioned over regions in the image where the object is not present and which can be traversed only with great difficulty by many optimisation algorithms. We have seen that the existence of a good background model can simplify the error surface to such an extent that we may only require a basic, local optimisation algorithm to effectively reach the global optimum. In the absence of such a comprehensive model for the background, the alternative is to use a powerful global optimisation approach. We have found that evolutionary methods such as SOMA and DE [Storn and Price (1997)] are very good candidates for handling the complicated error surfaces which are a common problem in template-based object recognition. In addition they require very little parameter configuration work making them applicable to many different problems and datasets. They are also flexible enough to cope with different types of variables, another characteristic of template matching applications. Another observation was that by allowing the global optimiser to execute for a limited number of function evaluations (or FEs for short) and switching to a local method when inside or near the basin of attraction of the global optimum, we can obtain results comparable to or better than those from a full, global optimisation run in a smaller amount of computation time.

Finally, from our research into Bayesian priors we found that the distribution of the scale parameter of an object imaged from random locations in an indoors environment seems to follow a lognormal model. Subsequent use of this model as a Bayesian prior can have better regularisation effects on the error function than an uninformative Gaussian distribution. More generally, we have discovered that a properly chosen Bayesian prior can help with the optimisation over the background regions especially when an explicit model is not available and at the same time assist in avoiding trivial solutions. Furthermore, it is preferable to restrict the variation of the solution parameters by penalising them according to a prior distribution than by explicitly setting boundaries in the optimisation algorithm configuration. This way we can focus the search on the interesting areas of the solution space while still maintaining a sufficiently high diversity in the search parameters.

8.2.1 Hypothesis 1

> *"It is possible to synthesise a novel view of an object and match it to a target image of that object. A good matching score will indicate that the object is present in the scene, and the object's pose and shape parameters are given by the LCV coefficients."*

Our initial implementation of the LCV approach (synthesis and matching steps in section 5.1) sup-

ported[3] the claim of our first hypothesis as we were able to use it and synthesise valid and realistic-looking views of the modelled object(s) taken from between the basis views. In addition, by using a matching function we managed to compare the synthesised image with the scene view and recover a matching score for different model configurations. Owing to the particular way this matching function was constructed (i.e. using prior distributions) a good score was only associated with a good match between the model and the object. We could then use this information together with the model's configuration and identify the location of the object in the scene image. The optimal configuration was also used in conjunction with the already recovered variation of the LCV coefficients partially to identify the pose of the scene object. In that way, the object's configuration is provided implicitly by the LCV coefficients. As a result, it was not possible to refute our first hypothesis, but have instead provided considerable evident to support it.

8.2.2 Hypothesis 2

> *" We can improve the accuracy and speed of the recovery of the model parameters of a rigid, 3-D object with the introduction of prior probability distributions in the template deformation process, based on previous knowledge of the underlying image generation process and imaging conditions."*

One of our principal speculations was that we could improve on a simple optimisation search over the solution space with the use of previously-known information about the objective function parameters by means of prior distributions. This information might be the variation, range or actual distributions of the individual parameters and result from the imaging conditions (e.g. sampling, camera parameters, light configurations and so on) that were used to generate the data.

Throughout our research (first in chapter 4 for 2-D and then in chapter 5 for 3-D) we demonstrated how it is possible to use such prior distributions to regularise the error surface, restrict the search to promising regions of the space and most importantly, avoid or remove any trivial solutions. In chapter 7 we had the opportunity to test this hypothesis with numerous experiments on real data using prior distributions based on the explicit knowledge about the variation of the LCV coefficients for the specific transformation of 3-D rotation about a vertical axis. We found that in all cases the priors resulted in a significant improvement in the performance (speed, efficiency and accuracy) of the search over standard maximum likelihood optimisation (or equally, using uninformative uniform priors) especially when the template model was positioned over background regions of the target image where, in the absence of a proper model the resulting objective function surface may be replete with many local minima, causing the optimisation algorithm to spend an unnecessarily large amount of time in these areas and possibly to converge incorrectly.

In the 2-D version studied first our priors were created to mimic the actual distributions of the transformation parameters (2-D affine) and we have acquired very good regularisation results and elimination of trivial solutions. This was illustrated by fast and accurate convergence to the global optimum (usually a minimum) using an elementary optimisation algorithm on real data and without the help of a

[3] Strictly speaking in scientific terms we have failed to falsify our hypothesis.

background model, although only a limited number of experiments were carried out since we required a simple proof of concept. In the 3-D version we instead used generic Gaussian priors (owing to the difficulty in decomposing or accurately composing the 3-D affine transformation matrix that implicitly incorporated the characteristics of the 10 LCV coefficients). We would have preferred to use the explicit models[4] but nevertheless the Gaussian alternatives proved to be very effective in capturing the underlying coefficient distribution.

Based on these results we have shown numerous times how such carefully chosen priors can assist the optimisation algorithm in accurately and efficiently recovering promising solutions, something which would otherwise very likely be difficult and time consuming no matter how elaborate the optimisation algorithm may be. The fact that we use subjective priors which include information about the LCV coefficients and also our expectation about the kind of transformation with which we are dealing we believe is more accurate and useful as far as the optimisation process is concerned rather than using the more objective, uninformative priors that are based only on the evidence observed during a run of the system. It may also be argued that from a Bayesian point of view the former approach (i.e. including information about the imaging process and conditions in the priors) is more valid since every available piece of information should be exploited accordingly. We may therefore claim that instead of refuting our second hypothesis, we have provided strong evidence supporting our original claims.

8.2.3 Hypothesis 3

> *"Recovery of the optimal LCV coefficients requires exhaustive search of the large solution space. By using an appropriate optimisation algorithm we can efficiently recover the optimal set of coefficients and thus recognise the object in the scene".*

We have already mentioned that in our work because of the type of features (intensity template) and the objective function used it is not possible to produce a closed-form solution to the object recognition problem. Instead we have to use an iterative optimisation approach to get as close as possible to the actual solution. Owing to the size and complexity (the presence of local optima, increased 'noisiness' of the objective function over background regions) of the solution space, the choice of a suitable optimiser is a very important factor in the performance of the recognition system. As part of our work into building a robust recognition system we investigated a number of different optimisation approaches, both traditional and some new to computer vision applications, global and local, stochastic and direct deterministic search. It soon became apparent that suitability could only be judged by considering the accuracy in terms of the error value reached and the efficiency in terms of the total number of objective function evaluations (or FEs for short). A local, direct search method is quite fast and efficient but suffers from a low accuracy (at least in our specific set-up). On the other hand, a global approach is slow but can recover more accurate results.

The natural progression was to combine the advantages of both methods in order to build a hybrid optimiser that is relatively efficient while retaining the accuracy associated with the global approach. This hybrid method was used throughout our 3-D object recognition experiments with very good results

[4] That may still be possible using the affine tri-focal tensor.

and detected the modelled objects in the scene in various configurations and under different imaging conditions. When compared to the other optimisation approaches, this method proved to be the preferred solution for our multi-view template matching problem. As a result we may assume that these observations provide adequate support for our third hypothesis.

8.2.4 Main hypothesis statement

> *"A solution to the view-based object recognition problem and the integration of the linear combination of views technique can be used to build a theoretical framework for the recognition of three-dimensional, rigid objects under a variety of configurations, using a small number of images taken from different viewpoints."*

The recognition system we have built, which combines the LCV theory for modelling the extrinsic variations in an object's appearance due to changes of viewpoint with the Bayesian framework and a powerful optimiser for recognising an object in an image is a realisation of our main hypothesis statement. The evidence we have provided so far in support for the three individual sub-hypotheses of this thesis when combined substantiate our main statement given above. Therefore we were unable to refute the main hypothesis and have in fact generated strong evidence in its favour.

Furthermore, by only finding support for our main hypothesis we have also achieved the main aim of this thesis which was to examine the suitability of the LCV theory for recognition of complicated objects using pixel intensity information. In addition, we managed to meet a number of the research objectives we set in section 1.3. More specifically, our system is capable of automatically detecting any (single instances of an) object in the scene without any manual intervention during the on-line search. A 3-D object may be modelled by using two or more basis views without any restrictions on its shape appearance and complexity. Furthermore, our tests showed that the system is relatively robust to noise and occlusion with little degradation in overall performance for moderate amounts of either and a predictable drop in efficiency when the noise or occluding surface are exaggerated.

We carried out a large array of tests on three public datasets with a combined number of 26 objects. Although this is by no means as large a number of objects as we originally hoped to use the system has nevertheless shown that it can handle the different scene configurations and that, no matter what the object shape complexity may be, the modelling process remains largely unaffected. Also, the identified variations of the LCV coefficient and the priors remained stable. In addition, the miss-match and false alarm errors were kept to a minimum as demonstrated by the well-defined diagonals in the model$\times$object or model$\times$pose arrays produced from our test experiments. Localisation mistakes were also low, especially in the absence of added noise or occlusion, with the modes of the test samples comfortably above the convergence thresholds and with acceptable recognition rates throughout all the datasets. It would have been desirable to execute additional experiments on other pose-variation datasets in order to get more general results, but however this has to be addressed in future work.

8.3 Limitations and future work

This work is not however without certain limitations which although they do not result in a deviation from the main scope of our research should at least be acknowledged in order to be addressed in future work. One of the more interesting topics for further investigation is the inclusion of intrinsic shape variations such as those that give rise to localised changes associated with facial expressions. As we have mentioned numerous times in this thesis, our approach only caters for extrinsic, pose variations that account only for global, 3-D affine deformations of the object of interest. By including the localised flexible changes we would be able to model and identify, for example, the expression of an individual together with the overall face shape and location in the scene.

For this to succeed however we will need to model the two different types of variation separately so they can be considered individually. The reason for this is so that we can define the object's implicit pose and shape configurations from the objective function parameters and possibly direct the search in each dimension based on each transformation's perceived characteristics, but most importantly in order to chose appropriate Bayesian priors for the independent, isolated deformations. Inclusion of intrinsic variations will allow us to deal with non-rigid 3-dimensional objects increasing thus the scope and applicability of our method.

To address this limitation, we may look at the work of Dias [Dias and Buxton (2002)] who managed to combine two flexible shape models (FSMs [Cootes et al. (2001)]) with a reformulation of the LCV theory and an alignment algorithm (Extended Procrustes Alignment - EPA) to create the integrated shape and pose model (ISPM). The ISPM does not mix the (intrinsic) shape and (extrinsic) pose variations as the two different types are modelled independently via the two component models (i.e. the multi-view FSM and the LCV), and it provides a better solution than the coupled-view FSM. Use of such an approach will require us however to re-evaluate our Bayesian prior models since in that work the LCV has been formulated using the central affine tri-focal tensor (CATT) and we would be dealing with additional variables and different types of transformations that may be difficult to bound and regularise. Furthermore, what we have learned about the characteristics of the error surfaces in template matching and about the 3-D affine transformations may be less relevant here because of the flexible shape deformations we will have to include.

We should note here that the ISPM method is a purely shape/feature driven approach that does not incorporate any texture information in the LCV or FSM models. In order to synthesise realistic-looking novel views and perform template-matching search on the ISPM it is necessary to include grey-scale information on this combined shape and pose model. This may be straightforward provided texture alignment can be achieved in a manner similar to the EPA algorithm. If texture alignment works then it might be possible to construct two flexible appearance models (FAMs [Cootes et al. (2001)]) and combine them into a multi-view IPAM (Integrated Pose and Appearance Model). However, it may be very complicated to build an equivalent alignment algorithm for implicitly transferring the intrinsic texture from an arbitrary image to the scene views. Despite all this, we believe that if one wishes to accurately and efficiently model flexible shape changes in an object, Dias' ISPM is a viable method to

consider as a starting point.

Another possible limitation is that both our LCV system and the ISPM model utilise only 2 basis views for the synthesis of novel images. This means that we can only deal with pose variations 'between' (or slightly outside) the angular range spanned by the viewpoints of the two basis views. If we wanted to work over a larger range of pose angles we would have to use additional 2-basis view models to capture the additional information. However, [Koufakis and Buxton (1998a); Kennedy et al. (1999); Buxton et al. (1998)] have shown that it is possible to use more than two views if necessary[5]. The questions that then need to be answered include: does an increase in the number of basis views and the pose angles they cover bring about a similar increase in the capacity of our model (i.e. can we synthesise and detect an object in this new, enlarged pose space); what is the maximum amount of joint-image space our model can include by adding new basis views, or in other words how many more basis views can we add to the system before we start to see no discernible increase in the pose angles we can model (law of diminishing returns); and is the model capacity controlled so that it remains sufficiently specific for object recognition. If we decide that there is not much practical advantage in using more than 2 basis views, then we could use several 2-view models and devise a switching scheme or selection process to work with the model that gives the best synthesis match.

So far we have seen that our LCV system is quite robust to the effects of occlusion. Although not examined from a strictly accurate statistical viewpoint, our limited tests have shown some initially positive results. What we have not studied in this thesis is the case of self-occlusions caused by non-convexities in the 3-D structure of the object. Such occlusions usually occur when we move to different regions of the view-sphere and cross over to a new scene aspect-view (regions over which small changes in viewpoint produce large changes of appearance [Koenderink and van Doorn (1979)] for which there is no equivalent in any of the basis views). In these cases, information necessary for synthesis and recognition are lost in the transition from basis views to scene view. Still however, it is possible to perform hidden surface removal by using the basis views to compute the affine depths [Koenderink and Doorn (1991)] at the control points of the basis images, similar to the work by [Hansard and Buxton (2000b)]. Since affine transformations are order-preserving we can use this information as input to a hidden surface removal program and resolve any self-occlusion ambiguities.

All our experiments so far have been limited to grey-scale images. This was done mainly for simplicity and speed since it is straightforward to extend the LCV synthesis step to colour images by applying the same process to each colour channel separately. Moreover, there are known forms of the cross-correlation measure applied to RGB images [Tsai et al. (2003)] and we may exploit this additional descriptive power in the three channels to assist with the optimisation search. The only possible problem we can anticipate at this stage is any artefacts that might arise due to our texture mapping/synthesis approach, especially for example at object boundary regions in the image and which can hinder the performance of the optimisation algorithm.

Throughout this research, we have demonstrated the positive effects of a proper background model.

[5] Reformulating the ISPM with more than two basis views via multi-view geometry will require higher-order multi-focal tensors.

We have examined a simplistic background model as an example in section 4.6 and have briefly noted the effects of having a known background in datasets like the CMU PIE [Sim et al. (2002)] (although not presented in this thesis in detail) and the Yale Face B, where the background image is wholly or partially provided. The goal of course would be to build a comprehensive statistical model of the background area in order to fully benefit in cases where it is not explicitly given with the rest of the data. A good starting point is perhaps the work by [Grenander and Srivastava (2001); Srivastava et al. (2002, 2003)] on the statistics of natural images and that of [Sullivan et al. (2001, 2000)] on foreground/ background mixture modelling. In fact, we have begun working on formalising a version of the LCV formulation with a basic, affine background intensity model that may be used as a stepping-stone to building more sophisticated algorithms. If we take the above one step further we may imagine also including an explicit model of the scene illumination and the non-affine changes in the scene (both foreground and background) photometry. Although we have observed that the current LCV formulation is able to cope with some lighting changes inclusion of a basic lighting model may be able to capture variations that the background model cannot deal with alone. We suggest looking at the models by [Georghiades et al. (2001)] that were developed with face recognition under varying illumination and pose in mind.

One of our future aims is to decompose the 3-D affine matrix as far as possible into individual, fundamental transformation not only for more efficient isolation and training of the LCV coefficients but also for a more statistically correct Bayesian formulation since strictly speaking the prior distribution if expressed as a product of separate distributions should correspond to independent variables. Furthermore, the LCV equations (3.14) need to be formulated by including the original constraints by [Ullman and Basri (1991)] and any constraints associated with the 3-D affine transformations since the linear system is over-complete with additional degrees of freedom [Buxton et al. (1998)]. The required decomposition or reformulation may not be possible with the affine matrix and so it might be necessary to consider the alternative route towards view-synthesis, which is using the affine tri-focal tensor [Shashua (1997)].

Ultimately, we would like to examine any possibilities into partially or fully automating the off-line landmark selection and correspondence establishment steps. At the moment, a relatively experienced user is required to choose a number of landmarks on prominent parts of the modelled object followed by establishing a valid correspondence in all the basis views. The long-term aim would be to make the system such that it can select the landmarks in all the basis images and establish the correspondences automatically. If such a feat is not possible we should at least allow for a non-expert user to pick out a set of landmarks independently in each image and perhaps determine an initial correspondence automatically. In order to ensure that the user has selected a useful set of landmarks the system might for example perform a few synthesis examples with ground truth data and calculate the match between synthesised and ground truth images. For this step, it is not necessary to have the LCV coefficient values but they can instead be interpolated based on the known variations (section 5.1.4) which are to a large extent, for most views, approximately object-independent.

Finally, it is our intention to carry out more experiments on additional datasets with significantly

more objects/individuals so as to get a better understanding of how our method performs in larger-scale classification problems and of the specificity of our models. The first one is the M2VTS multi-modal face database by [Pigeon and Vandendorpe (1997)] which contains images of 37 individuals across $\pm 90^0$ pose variation, with localised face changes (and in specific lip movement) and the existence of facial accessories such as glasses, scarves etc. The other dataset is part of the Face Recognition Grand Challenge (FRGC) [Phillips et al. (2005)] and includes training and validation subsets of frontal images of various individuals, each images across two facial expressions and in both controlled and varying illumination settings. Both of these databases have been used extensively for the evaluation of object (face) detection algorithms, and as a result our experiments can be compared with recent, competing methods.

Furthermore, we would like to research on possible ways of improving the execution speed of the search, perhaps by reducing the time required for a single synthesis (which equals one FE). One possibility is to make use of the latest dedicated graphics hardware and map the synthesis straight onto the GPU or by using the standard graphics APIs [Hansard and Buxton (2000a)].

Appendices

Appendix A

Algorithms

In this section we include some more details, in the form of pseudocode, on the various algorithms presented and used in this thesis.

Algorithm 1 Restarting simplex algorithm pseudocode

```
while iterations k<N do
  Rank simplex vertices  //(Best, Worst, nextWorst)
  R=Reflect(Worst);  //Make a reflection R
  if R<Best then
    E=Expand(Worst);  //Make expansion E
    if E<R then
      Worst=E  //Replace worst point with E
    else
      Worst=R  //Replace worst point with R
    end if
  else if R<nextWorst then
    Worst=R  //Replace worst point with R
  else if R<Worst then
    C_p=posContract(Worst)   //Make a positive contraction C_p
    Worst=C_p  //Replace worst point with C_p
  else
    C_n=negContract(Worst)  //Make a negative contraction C_n
    Worst=C_n  //Replace worst point with C_n
  end if
  if Simplex has stalled then
    Restart simplex
  end if
end while
```

Algorithm 2 A general pattern search algorithm pseudo-code.

```
for iterations k = 0, 1, ... do
  Compute function at f(x)
  Determine a step s_k using exploratory moves algorithm
  if f(x_k) < f(x_k + s_k) then  then
    x_{k+1} = x_k + s_k
  else
    x_{k+1} = x_k
  end if
  Update C_k and Δ_k
end for
```

Algorithm 3 Genetic algorithm pseudocode

```
Set g = 0   //generation counter
Initialise population P(g)
Evaluate population P(g)  //compute fitness values
repeat
  g = g + 1
  Select P(g) from P(g − 1)  //perform competitive selection
  Crossover population P(g)
  Mutate population P(g)
  Evaluate population P(g) //compute fitness values
until terminating condition
```

Algorithm 4 Differential evolution pseudocode

```
for each target x⃗_{i,G} vector in current generation G do
  Randomly choose two population members x⃗_{r1,G} and x⃗_{r2,G}
  Build weighted difference vector x⃗_{r1,2,G} = F(x⃗_{r1,G}, x⃗_{r2,G})
  Add a third randomly chosen vector x⃗'_{i,G} = x⃗_{r1,2,G} + x⃗_{r3,G}
  Crossover with target vector u⃗_{i,G+1} = x⃗'_{i,G} ⊗ x⃗_{i,G}
  if f(u⃗_{i,G+1}) < f(x⃗_{i,G}) then
    x⃗_{i,G+1} = u⃗_{i,G+1}
  else
    x⃗_{i,G+1} = x⃗_{i,G}
  end if
end for
```

Algorithm 5 SOMA pseudocode

```
Generate new random population within bounds.
Find index of leader L
for each migration do
  for each individual in population do
    for each step in pathLength do
      Generate new PRTVector for the individual
      Calculate new position x⃗ = x⃗_0 + m⃗ t PRTVector
      if f(x⃗) < f(x⃗_0)x then
        Accept x⃗
      end if
    end for
  end for
  Find index of leader L
end for
```

Algorithm 6 AAM search single iteration

```
Evaluate the difference δ_{g0} = g_{s0} − g_{m0} between the model's graylevels
and the image sample g_s.
Evaluate the error E_0=|δ_{g0}|^2
Compute the predicted displacement δ_c = AΔ.
Set k=1
Let c_1=c_0 - kδ_c
Sample the image at this new configuration and calculate E_1 = |δ_{g1}|^2 =
|g_{s1} − g_{m1}|^2
if E_0>E_1 then
  Accept new configuration at c_1
else
  Try at k=1.5, 0.5, ...
end if
```

Appendix B

Exploratory data analysis techniques

Quantitative techniques take all of the data and map it into a few numbers describing the modelling process and the parameter estimates. The advantage of such methods is that these few numbers focus on important trends (location, variation and so on) of the population while being sensitive to any changes in that data (for example shift in location). However overly concentrating on these few properties can filter out other important characteristics such as skewness, tail length, autocorrelation and so on. Graphical methods on the other hand make use of all the available data and present information in such a way that combined with our natural pattern-recognition abilities they allow us to gain additional insight into the data.

We present the following standard graphical methods: a probability plot, a histogram with overlaid estimated parametric pdf, and an empirical cumulative distribution function (cdf) with overlaid estimated parametric cdf. A probability plot [Chambers et al. (1983)] is a graphical technique for qualitatively assessing the fit of data to a theoretical distribution. In this plot the data is drawn against a theoretical distribution in such a way that the points should lie approximately on a straight line. Departures from this straight line indicate departures from the distribution. Suppose that we have ordered sample values $X_i = X_1, X_2, \ldots, X_N$, called *order statistics*, and the hypothesis that X_i follows a certain distribution F. The probability plot is formed by plotting:

$$X_i \text{ vs. } F^{-1}\left(\frac{i}{N+1}\right) \tag{B.1}$$

where F^{-1} is the percent point function (inverse of the cdf) of the hypothesised distribution. The pdf and cdf are obtained by maximum likelihood estimation (MLE). Given N ordered data points $X_i = X_1, X_2, \ldots, X_N$ the empirical cdf is defined as:

$$E_N = \frac{n(i)}{N} \tag{B.2}$$

where $n(i)$ is the number of points less than X_i. This essentially is a step function that increases by $1/N$ at the value of each ordered point. The larger the sample size the smaller the increase step and thus the closer the estimated empirical cdf matches the actual cdf.

In addition we introduce the following quantitative methods: the Kolmogorov-Smirnov (K-S) test

[Chakravarti et al. (1967)] and the Anderson-Darling (A-D) test [Stephens (1974)]. The K-S test is used to decide if a sample comes from a population with a specific distribution and is based on the empirical distribution function. It depends on the maximum difference between a hypothesised theoretical distribution and the empirical distribution. More rigorously, the K-S test is defined by two hypotheses H_0 and H_1, the test statistic, the significance level α and the critical region. The simple, null hypothesis H_0 states that the data follows a specified distribution, and conversely the alternate hypothesis H_1 states that the data does not follow the specified distribution. The test statistic is defined as:

$$D = \max_{1 \leq i \leq N} \left| F(X_i) - \frac{i}{N} \right|, \tag{B.3}$$

where F is the theoretical cdf of the distribution being tested which must be continuous and fully specified. The significance level is the probability of rejecting the null hypothesis when it is in fact true. Finally, the critical region may be obtained from statistical tables depending on the significance level and the hypothesis H_0 is rejected if D is greater than a given critical value.

The A-D test is a modification of the K-S test that gives more weight to the tails of the distribution. Although the K-S test is distribution-free, in the sense that its critical values are not dependent on a specific distribution, the A-D test makes use of specific distributions in calculating critical values. The advantage of this is that it allows for a more sensitive test but on the other hand critical values must be calculated for each distribution and unfortunately we were unable to find critical value tables in the literature for some of the distributions. The A-D test statistic is defined as:

$$A^2 = -N - S \tag{B.4}$$

where

$$S = \sum_{i=1}^{N} \frac{2i-1}{N} [\ln F(X_i) + ln(1 - F(X_{N+1-i}))] \tag{B.5}$$

and N, F and X_i are as above. For a given distribution the A-D test may be multiplied by a factor dependent on the sample size N. We call this the "adjusted A-D" statistic and this is what should be compared against the critical values.

Bibliography

Amit, Y., Grenander, U. and Piccioni, M. (1991), Structural image restoration through deformable templates, *Jour. of the American Statistical Assosiation* **86**(414), 376–387.

Amit, Y. and Trouve, A. (2007), POP: Patchwork of Parts Models for Object Recognition, *IJCV* **75**(2), 267–282.

Audet, C. and Jr., J. E. D. (2003), Analysis of Generalized Pattern Searches, *SIAM Journal on Optimization* **13**(3), 889–903.

Ballard, D. H. (1981), Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition* **13**, 111.

Bebis, G., Louis, S., Varol, T. and Yfantis, A. (2002), Genetic Object Recognition Using Combinations of Views, *IEEE Transactions on Evolutionary Computation* **6**(2), 132–146.

Bebis, G., Uthiram, S. and Georgiopoulos, M. (1999), Genetic search for face detection and verification, *in International Conference on Information Intelligence and Systems*, pp.360–367.

Beichel, R., Bischof, H., Leberl, F. and Sonka, M. (2005), Robust active appearance models and their application to medical image analysis, *IEEE Transactions on Medical Imaging* **24**(9), 1151–1169.

Bergevin, R. and Levine, M. D. (1993), Generic object recognition: Building and matching coarse descriptions fromlinedrawings, *IEEE Trans. Patt. Anal. Machine Intell.* **15**(1), 19–36.

Besl, P. J. and Jain, R. C. (1985), Three-Dimensional Object Recognition, *ACM Computing Surveys (CSUR)* **17**, 75–145.

Betke, M. and Makris, N. C. (1995), Fast object recognition in noisy images using Simulated Annealing, *in Proceedings of the Fifth International Conference on Computer Vision*, pp.523–530.

Biederman, I. (1993), Geon Based Object Recognition, *in BMVC*.

Binford, T. and Levitt, T. (1996), Model-based recognition of objects in complex scenes, *Image understanding workshop* pp.89–100.

Bishop, C. M. (1995), *Neural Networks for Pattern Recognition.*, Oxford University Press.

Blake, A. and Isard, M. (1998), Active Contours, *Springer* .

Blanz, V., Scholkopf, B., Büllthoff, H., Burges, C. and andT.Vetter, V. V. (1996), Comparison of view-based object recognition algorithms using realistic 3Dmodels, *Artificial Neural Networks ICANN'96. Springer Lecture Notes in ComputerScience* **1112**, 251–256.

Borotschnig, H., Paletta, L., Prantl, M. and Pinz, A. (2000), Appearance-based active object recognition, *Image and Vision Computing* **18**, 715–727.

Brown, L. G. (1992), A survey of image registration techniques, *ACM Computing Surveys* **24**(4), 325–376.

Brunelli, R. and Poggio, T. (1993), Face Recognition: Features versus Templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(10), 1042–1052.

Bülthoff, H. and Edelman, S. (1992), Psychophysical support for a two-dimensional view interpolation theory of object recognition, *Proc. Natl. Acad. Sci. USA* **89**, 60–64.

Burr, D. J. (1981), Elastic matching of line drawings, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **3**(6), 708–713.

Buxton, B. (2004), private communication.

Buxton, B. F., Shafi, Z. and Gilby, J. (1998), Evaluation of the construction of novel views by a combination of basis views., *in Proc. IX European Signal Processing Conference (EUSIPCO-98)*, Rhodes, Greece.

Buxton, B. and Zografos, V. (2005), Flexible template and model matching using image intensity, *in Proceedings Digital Image Computing: Techniques and Applications (DICTA)*, pp.438 – 447.

Caelli, T. and Kosinov, S. (2004), An Eigenspace Projection Clustering Method for Inexact Graph Matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(4), 515–519.

Califano, A. and Mohan, R. (1994), Multidimensional indexing for recognizing visualshapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**(4), 373–392.

Canny, J. F. (1986), A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(6), 679–698.

Chakravarti, I. M., Laha, R. G. and Roy, J. (1967), *Handbook of Methods of Applied Statistics*, Vol. I, John Wiley and Sons, pp.392–394.

Chambers, J., Cleveland, W., Kleiner, B. and Tukey, P. (1983), *Graphical Methods for Data Analysis*, Chapman & Hall.

Cho, K.-S. and Kim, Y.-G. (2007), *Universal Access in Human-Computer Interaction. Ambient Interaction (LNCS)*, Vol. 4555, Springer Berlin / Heidelberg, Chapter Continuous Recognition of Human Facial Expressions Using Active Appearance Model, pp.777–783.

Christensen, G. E., Rabbitt, R. D. and Miller, M. I. (1996), Deformable templates using large deformation kinematics, *IEEE Trans. on Image Processing* **4**(10), 1435–1447.

Cipolla, R. and Blake, A. (1990), The dynamic analysis of apparent contours, *Proc. 3th Int. Conf. on Computer Vision* pp.616–623.

Clemens, D. and Jacobs, D. (1991), Space and time bounds on indexing 3d models from 2d images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(10), 1007–1017.

Cootes, T. F., Edwards, G. J. and Taylor, C. J. (2001), Active Appearance Models, *IEEE Pattern Analysis and Machine Intelligence* **23**, 681–685.

Cootes, T. F. and Taylor, C. J. (2004), Statistical Models of Appearance for Computer Vision, Technical Report, University of Manchester.

Cootes, T. F., Taylor, C. J., Cooper, D. H. and Graham, J. (1995), Active Shape Models - their training and application, *Computer Vision and Image Understanding* **61**, 38–59.

Cootes, T. F., Wheeler, G. V., Walker, K. N. and Taylor, C. J. (2000), Coupled-View Active Appearance Models, *Proc. BMVC* **1**, 52–61.

Cox, I. J., Rao, S. B. and Zhong, Y. (1996), Ratio regions a technique for image segmentation, *International Conference on Pattern Recognition* pp.557–564.

Dantzig, G. B. (1963), *Linear Programming and Extensions.*, Princeton University.

Darell, T., Gordon, G., Harville, M. and Woodfill, J. (2000), Integrated Person Tracking Using Stereo, Color and Pattern Detection, *Int. Journal of Computer Vision* **37**(2), 175–185.

DeJong, K. A. (1975), An analysis of the behavior of a class of genetic adaptive systems., PhD thesis, University of Michigan, Ann Arbor, MI, USA.

Delaunay, B. (1934), Sur la sphre vide, *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk* **7**, 793–800.

Dias, B. (2004), Implicit, View-Invariant Modelling of 3D Non-Rigid Objects, PhD thesis, University College London.

Dias, B. and Buxton, B. F. (2002), Integrated Shape and Pose Modelling, *Proceedings of the British Machine Vision Conference (BMVC 2002)* pp.827–836.

Dong, J.-X., Kryzak, A. and Suen, C. Y. (2005), Fast SVM Training Algorithm with Decomposition on Very Large Data Sets, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(4), 603–618.

Duta, N. and Sonka, M. (1998), Segmentation and interpretation of MR brain images: an improved active shape model, *IEEE Transactions on Medical Imaging* **17**(6), 1049–1062.

Edwards, G., Taylor, C. J. and Cootes, T. F. (1998), Interpreting Face Images Using Active Appearance Models, *in AFGR98*, pp.300–305.

Evans, M., Hastings, N. and Peacock, B. (2002), *Statistical Distributions*, third edition, Wiley Series in Probability and Statistics.

Faugeras, O. D. and Hebert, M. (1983), A 3-D Recognition and Positioning Algorithm Using Geometrical Matching BetweenPrimitiveSurfaces, *in Proc. 8th Int. Joint Conf. Artificial Intell.*, pp.996–1002.

Felzenswalb, P. (2005), Representation and detection of deformable shapes, *IEEE PAMI* **27**(2), 208–220.

Fergus, R., Perona, P. and Zisserman, A. (2003), Object class recognition by unsupervised scale-invariant learning, *Computer Vision and Pattern Recognition* **2**, 264–271.

Figueiredo, M., Leitao, J. and Jain, A. K. (1997), Adaptive B-splines and boundary estimation, *Proceedings of CVPR '97* pp.724–730.

Fischler, M. A. and Bolles., R. C. (1981), Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography., *Comm. of the ACM* **24**, 381–395.

Fua, P. and Brechbuhler, C. (1996), Imposing hard constraints on soft snakes, *Proc. European Conference on Computer Vision* pp.495–506.

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (1995), *Bayesian Data Analysis*, 2nd edition, Chapman and Hall, London.

Georghiades, A. S., Belhumeur, P. N. and Kriegman, D. J. (2001), From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose, *IEEE Trans. Pattern Anal. Mach. Intelligence* **23**(6), 643–660.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimzation & Machine Learning*, Addison-Wesley.

Gong, S., Psarrou, A. and Romdhani, S. (2002), Corresponding dynamic appearances, *Image Vision Comput.* **20**, 289–300.

Goodall, C. (1991), Procrustes methods in the statistical analysis of shape., *Journal of the Royal Statistical Society* **53**(2 of Series B), 285–339.

Goshtasby, A. (1986), Piecewise Linear Mapping Functions for Image Registration, **19**(6), 459–466.

Gower, J. C. (1975), Generalized procrustes analysis, *Psychometrika* **40**, 33–51.

Gradshteyn, I. S. and Ryzhik, I. M. (1980), *Table of Integrals, Series and Products*, 4th edition, Academic Press.

Grenander, U. (1993), *General Pattern Theory: A Mathematical Study of Regular Structures*, Oxford University Press.

Grenander, U., Chow, Y. and Keenan, D. M. (1991), Hands: A Pattern Theoretic Study of Biological Shapes, *Springer* .

Grenander, U. and Keenan, D. M. (1993), Towards automated image understanding, *in* K. V. Mardia and G. K. Kanji (eds.), *Advances in Applied Statistics: Statistics and Images*, Carfax Publishing Company, pp.89–103.

Grenander, U. and Srivastava, A. (2001), Probability Models for Clutter in Natural Images, *IEEE PAMI* **23**(4), 424–429.

Grimson, W. E. L. (1990), *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press.

Grimson, W. E. L. and Huttenlocher, D. P. (1988), On the Sensitivity of the Hough Transform for Object Recognition, Technical Report, MIT Artificial Intelligence Laboratory.

Grimson, W. and Huttenlocher, D. (1990), On the sensitivity of geometric hashing, *Proc. ICCV* pp.334–338.

Grimson, W. and Lozano-Perez, T. (1986), Model-based recognition and localization from tactile data, *Journal of Robotics Research* **3**(3), 3–35.

Gross, R., Matthews, I. and Baker, S. (2004), Appearance-Based Face Recognition and Light-Fields, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(4), 449–465.

Guitton, A. and Symes, W. W. (2003), Robust inversion of seismic data using the Huber norm, *Geophysics* **68**(4), 1310–1319.

Guitton, A. and Verschuur, D. J. (2004), Adaptive subtraction of multiples using the L1-norm, *Geophysical Prospecting* **52**, 27–38.

Hansard, M. E. and Buxton, B. F. (2000a), Image-based rendering via the standard graphics pipeline, *IEEE International Conference on Multimedia and Expo* **3**, 1437–1440.

Hansard, M. E. and Buxton, B. F. (2000b), Parametric view-synthesis, *In Proc. 6th ECCV* **1**, 191–202.

Harris, C. G. and Stephens, M. (1988), A combined corner and edge detector., *in In 4th Alvey Vision Conference*, pp.147–151.

Hasegawa, O. and Kanade, T. (2005), Type classification, color estimation, and specific target detection of movingtargets on public streets., *Machine Vision and Applications* **16**(2), 116–121.

Hastings, W. (1970), Monte Carlo Sampling Methods Using Markov Chains and Their Applications, *Biometrika* **57**(1), 97–109.

Hemayed, E. E. (2003), A survey of camera self-calibration, *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03)* pp.351–357.

Hill, A., Taylor, C. J. and Cootes, T. F. (1992), Object Recognition by Flexible Template Matching using Genetic Algorithms, *in Proceedings of the Second European Conference on Computer Vision*, Springer-Verlag, London, UK, pp.852–856.

Hill, D. L. G., Batchelor, P. G., Holden, M. and Hawkes, D. J. (2001), Medical Image Registration [invited topical review], *Physics in Medicine and Biology* **46**(3), R1–R45.

Holland, J. H. (1992), *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.*, The MIT Press.

Huang, C., Camps, O. and Kanugo, T. (1997), Object recognition using appearance-based parts and relations, *Computer Vision and Pattern Recognition Conference* pp.887–883.

Huang, J. and Mumford, D. (1999), Statistics of Natural Images and Models, *Computer Vision and Pattern Recognition* **1**, 1541–1547.

Huber, P. J. (1973), Robust regression: Asymptotics, Conjectures and Monte Carlo, *The Annals of Statistics* **1**, 799–821.

Huttenlocher, D. and Ullman, S. (1990), Recognizing solid objects by alignment with an image, *International Journal of Computer Vision* **5**(2), 195–212.

Isard, M. and Blake, A. (1998), CONDENSATION-conditional density propagation for visual tracking, *Int. Journal of Computer Vision* **29**, 5–28.

Jacobs, D. (1997), Matching 3d models to 2d images, *International Journal of Computer Vision* **21**, 123–153.

Jain, A. K., Zhong, Y. and Dubuisson-Jolly, M.-P. (1998), Deformable template models: A review, *Signal Processing* **71**(2), 109–129.

Jain, A. K., Zhong, Y. and Lakshmanan, S. (1996), Object Matching Using Deformable Templates, *IEEE PAMI* **18**(3), 267–278.

Jaynes, E. T. (2003), *Probability Theory : The Logic of Science*, Cambridge University Press.

Joliffe, T. (1986), *Principal Components Analysis*, Springer-Verlag.

Jolly, M.-P. D., Lakshmanan, S. and Jain, A. K. (1996), Vehicle segmentation and classification using deformable templates, *IEEE Trans. Pattern Analysis and Machine Intelligence* **18**(3), 293–308.

Kass, M., Witkin, A. and Terzopoulos, D. (1988), Snakes: active contour models, *International Journal of Computer Vision* **1**(4), 321–331.

Kennedy, D. M., Buxton, B. F. and Gibly, J. H. (1999), Application of the total least squares procedure to linear view interpolation, *Proc. BMVC* **1**, 305–314.

Kim, H.-D., Park, C.-H., Yang, H.-C. and Sim, K.-B. (2006), Genetic Algorithm Based Feature Selection Method Development for Pattern Recognition, *in International Joint Conference SICE-ICASE*.

Kim, S.-H., Kim, N.-K., Ahn, S. C. and Kim, H.-G. (1998), Object Oriented Face Detection Using Range and Color information, *Proc. Third International Conference on Automatic Face and Gesture Recognition* pp.76–81.

Klein, A. K., Lee, F. and Amini, A. A. (1997), Quantitative coronary angiography with deformable spline models, *IEEE Trans. on Medical Imaging* **16**(5), 468–482.

Koenderink, J. and van Doorn, A. (1979), The internal representation of solid shape with respect to vision, *Biological Cybernetics* **32**, 211–216.

Koenderink, J. J. (1990), *Solid Shape*, MIT Press, Cambridge, MA.

Koenderink, J. J. and Doorn, A. J. V. (1991), Affine structure from motion, *Journal of the Optical Society of America* **8**, 377–385.

Kohavi, R. (1995), A study of cross-validation and bootstrap for accuracy estimation and model selection, *in Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Vol. 2, Morgan Kaufmann, San Mateo, pp.1137–1143.

Koufakis, I. and Buxton, B. (1998a), Linear Combination of Face Views for Low Bit Rate Face Video Compression, *in Proceedings of the IX European Signal Processing Conference (EUSIPCO-98)*, Rhodes, Greece, pp.2305–2308.

Koufakis, I. and Buxton, B. F. (1998b), Very low bit-rate face video compression using linear combination of 2d face views and principal components analysis, *Image and Vision Computing* **17**, 1031–1051.

K.Shoemake and Duff, T. (1992), Matrix Animation and Polar Decomposition, *Proceedings of Graphics Interface* pp.258–264.

Lakshmanan, S., Jain, A. K. and Zhong, Y. (1995), Detecting straight edges in millimeter wave images, *Proc. International Conference on Image Processing* pp.258–261.

Lamdan, Y. and Wolfson, H. J. (1988), Geometric Hashing: A General and Efficient Model-Based Recognition Scheme, *Proceedings Second International Conference on Computer Vision* pp.238–249.

Lamdan, Y., Schwartz, J. and Wolfson, H. (1988), Object recognition by affine invariant matching, *in Computer Society Conference on Computer Vision and Pattern Recognition CVPR '88*, pp.335–344.

Larsen, R. and Eiriksson, H. (2002), L1 Generalized Procrustes 2D Shape Alignment, *in Eleventh International Workshop on Matrices and Statistics, Informatics and Mathematical Modelling*, Techincal University of Denmark.

Lee, M. W. and Ranganath, S. (2003), Pose-invariant face recognition using a 3D deformable model, *Pattern Recognition* **36**, 1835–1846.

Leinhardt, G. and Leinhardt, S. (1980), *Exploratory Data Analysis: New Tools for the Analysis of Empirical Data. Review of Research in Education*, Vol. 8.

Leung, T. K., Burl, M. C. and Perona, P. (1998), Probabilistic Affine Invariants for Recognition, *Proc. of the IEEE Conf. on Comp. Vision and Pattern Recognition* pp.678–684.

Levenberg, K. (1944), A Method for the Solution of Certain Problems in Least Squares, *Quart. Appl. Math.* **2**, 164–168.

Leventon, M. E., Grimson, W. E. L. and Faugeras, O. (2000), Statistical shape influence in geodesic active contours, *Proc. Conf. on Computer Vision and Pattern Recognition* **I**, 316–323.

Li, Y., Gong, S. and H.Liddell (2000), Support vector regression and classification based multi-view face detection and recognition, *Proc. AFGR* pp.300–305.

Li, Y., Gong, S., Sherrah, J. and Liddell, H. (2004), Support Vector Machine based multi-view face detection and recognition, *Image and Vision Computing, 22(5), pages 413-427, 2004.* **22**(5), 413–427.

Liang, D., Yang, J., Zheng, Z. and Chang, Y. (2005), A facial expression recognition system based on supervised locally linear embedding, *Pattern Recognition Letters* **26**(15), 2374–2389.

Liang, L., Wen, F., Tang, X. and Xu, Y.-Q. (2006), An Integrated Model for Accurate Shape Alignment, *in ECCV*, pp.333–346.

Loizides, A., Slater, M. and Langdon, W. B. (2001), Measuring facial emotional expressions using genetic programming., *Soft computing and industry recent applications* pp.545–554.

Lowe, D. (1985), *Perceptual Organization and Visual Recognition*, Kluwer Academic Publishers, Boston MA.

Lowe, D. G. (1987), Three-Dimensional Object Recognition from Single Two-Dimensional Images, *Artificial Intelligence* **31**(3), 355–395.

Maes, F., Vandermeulen, D. and Suetens, P. (1999), Comparative evaluation of multiresolution optimization strategies for multimodality image registration by maximization of mutual information, *Medical Image Analysis* **3**(4), 373–386.

Marcini, D., Shokoufandeh, A., Dickinson, S., Siddiqi, K. and Zucker, S. (2002), View-based 3d object recognition using shock graphs, *Proceedings 16th International Conference on Pattern Recognition* **3**, 24–28.

Marquardt, D. (1963), An Algorithm for Least-Squares Estimation of Nonlinear Parameters, *SIAM J. Appl. Math.* **11**, 431–441.

Marr, D. (1982), *Vision: a computational investigation into the human representation andprocessingof visual information*, W. H. Freeman.

Matthews, I. and Baker, S. (2004), Active Appearance Models Revisited, *International Journal of Computer Vision* **60**(3), 135–164.

Maybank, S. J. (1998), Relation between 3D invariants and 2D invariants, *Imaging and Vision Computing* **16**, 13–20.

Mekuz, N., Bauckhage, C. and Tsotsos, J. K. (2005), Face Recognition with Weighted Locally Linear Embedding, *in Proceedings of Canadian Conference on Computer and Robot Vision*, pp.290–296.

Menet, S., Saint-Marc, P. and Medioni, G. (1990), B-snakes: implementation and application to stereo, *Proc. DARPA* pp.720–726.

Metropolis, A., Rosenbluth, W., Rosenbluth, M., Teller, H. and Teller, E. (1953), Equation of state calculations by fast computing machines, *J. Chem. Phys.* **21**(6), 1087–1092.

Mitchell, S. C., Lelieveldt, B. P. F., van der Geest, R. J., Bosch, H. G., Reiver, J. H. C. and Sonka, M. (2001), Multistage hybrid active appearance model matching: segmentation of left and right ventricles in cardiac MR images, *IEEE Transactions on Medical Imaging* (20), 415–423.

Mumford, D. (1996), The statistical description of visual signals, *in* O. Mahrenholtz and R. Mennicken (eds.), *ICIAM 95*, Verlag.

Murase, H. and S.Nayar (1995), Visual learning and recognition of 3-d objects from appearance, *International journal of computer vision* **14**, 5–24.

Nelder, J. A. and Mead, R. (1965), A simplex method for function minimization, *Computer Journal* **7**, 308–313.

Nene, S. A., Nayar, S. K. and Murase, H. (1996), Columbia Object Image Library (COIL-20), Technical Report CUCS-006-96, Department of computer science, Columbia University, New York, N.Y. 10027.

Neumann, A. and Lorenz, C. (1998), Statistical shape model based segmentation of medical images, *Computerized Medical Imaging and Graphics* **22**(2), 133–143.

Ng, J. and Gong, S. (1999), Multi-view face detection and pose estimation using a composite support-vectormachine across the view sphere, *IEEE International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems* pp.14–21.

Nocedal, J. and Wright, S. (1999), *Numerical optimization*, Springer, New York.

Nolle, L., Zelinka, I., Hopgood, A. A. and Goodyear, A. (2005), Comparison of a self-organizing migration algorithm with simulated annealing and differential evolution for automated waveform tuning., *Advances in Engineering Software* pp.1–9.

Ohba, K. and Ikeuchi, K. (1997), Detectability, uniqueness and reliability of eigenwindows for stable verificationofpartially occluded objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **9**(9), 1043–1048.

Olson, C. F. (2002), Maximum-Likelihood Image Matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(6), 853–857.

Oplatkova, Z. and Zelinka, I. (2006), Investigation on Artificial Ant using Analytic Programming, *in GECCO*, Seatle, WA, USA.

Osuna, E., Freund, R. and Girosi, F. (1997), Training Support Vector Machines: An application to Face Detection, *Proc. IEEE Conference on Computer Vision and Pattern Recognition* pp.130–136.

Papageorgiou, C. and Poggio, T. (2000), A Trainable System for Object Recognition, *Int. Journal of Computer Vision* **38**(1), 15–33.

Paragios, N. and Deriche, R. (2000), Geodesic active contours and level sets for the detection and tracking ofmovingobjects, *IEEE Trans. on Pattern Recognition and Machine Intelligence* **22**(3), 266–280.

Parke, F. I. and Waters, K. (1996), *Computer Facial Animation*, A. K. Peters Ltd, MA.

Peters, G. (2000), Theories of Three-Dimensional Object Perception: A Survey,, *Recent Research Developments in Pattern Recognition, Transworld ResearchNetwork* .

Peters, G. and von der Malsburg, C. (2001), View Reconstruction by Linear Combination of Sample Views, *In Proc. British Machine Vision Conference BMVC 2001* **1**, 223–232.

Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., Chang, J., Hoffman, K., Marques, J., Min, J. and Worek, W. (2005), Overview of the Face Recognition Grand Challenge, *in IEEE Conference on Computer Vision and Pattern Recognition*, pp.947–954.

Pigeon, S. and Vandendorpe, L. (1997), The M2VTS Multimodal Face Database (Release 1.00), *in Proceedings of the First International Conference on Audio- and Video-Based Biometric Person Authentication, LNCS*, Vol. 1206, Springer-Verlag, pp.403 – 409.

Poggio, T. and Edelman, S. (1990), A Network that Learns to Recognize Three Dimensional Objects, *Nature* **343**, 263–266.

Pontil, M. and Verri, A. (1998), Support vector machines for 3-d object recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(6), 637–646.

Pope, A. R. (1994), Model-based object recognition. A survey of recent research, Technical Report 94-04.

Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T. (1993), *Numerical Recipes in C The Art of Scientific Computing*, Cambridge University Press.

Revaud, J., Lavoue, G., Ariki, Y. and Baskurt, A. (2007), Fast and cheap object recognition by linear combination of views., *Proceedings of the 6th ACM international conference on Image and video retrieval* pp.194–201.

Roweis, S. and Saul, L. (2000), Nonlinear dimensionality reduction by locally linear embedding., *Science* **290**(5500), 2323–2326.

Salvi, J., Armangu, X. and Batlle, J. (2002), A comparative review of camera calibrating methods with accuracy evaluation, *Pattern Recognition* **35**(7), 1617–1635.

Schmid, C. and Mohr, R. (1997), Local grayvalue invariants for image retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**, 530–534.

Scholkopf, B., Smola, A. and Miller, K. (1998), Nonlinear component analysis as a kernel eigenvalue problem., *Neural Computation* **10**(5), 1299–1319.

Sclaroff, S. and Isidoro, J. (1998), Actibe blobs, *Proc. 6th Int. Conf. on Computer Vision* pp.1146–1153.

Sebe, N. and Lew, M. S. (2001), Maximum Likelihood Stereo Matching, *ICPR01* **1**, 900–903.

Sebe, N. and Lew, M. S. (2002), Maximum Likelihood Shape Matching, *ACCV2002: The 5th Asian Conference on Computer Vision* pp.713–718.

Sethian, J. A. (1999), *Level Set Methods and Fast Marching Methods*, Cambridge Monographs on Applied and Computational Mathematics (No. 3), 2 edition, Cambridge University Press.

Shashua, A. (1992), Geometry and Photometry in 3D Visual Recognition, PhD thesis, Massachusetts Institute of Technology.

Shashua, A. (1997), Trilinear Tensor: The Fundamental Construct of Multiple-view Geometry and Its Applications, *in AFPAC '97: Proceedings of the International Workshop on Algebraic Frames for the Perception-Action Cycle*, Springer-Verlag, London, UK, pp.190–206.

Shewchuk, J. R. (2002), Delaunay Refinement Algorithms for Triangular Mesh Generation, *Computational Geometry: Theory and Applications* **22**, 21–74.

Siddiqi, K., Shokoufandeh, A., Dickinson, S. J. and Zucker, S. W. (1999), Shock Graphs and Shape Matching, *International Journal of Computer Vision* **35**(1), 13–32.

Sim, T., Baker, S. and Bsat, M. (2002), The CMU pose, illumination and expression (PIE) database, *in Proc. of the 5th IEEE international conference on automatic face and gesture recognition*.

Srivastava, A., Lee, A., Simoncelli, E. and Zhu, S.-C. (2003), On Advances in Statistical Modeling of Natural Images, *Journal of Mathematical Imaging and Vision* **18**, 17–33.

Srivastava, A., Liu, X. and Grenander, U. (2002), Universal Analytical Forms for Modeling Image Probabilities, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(9), 1200–1214.

Staib, L. H. and Duncan, S. (1992), Boundary finding with parametrically deformable models, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **14**(11), 1061–1075.

Stephens, M. A. (1974), EDF Statistics for Goodness of Fit and Some Comparisons, *Journal of the American Statistical Association* **69**, 730–737.

Stommel, M. and Kuhnert, K.-D. (2009), A Hierarchical Model for the Recognition of Deformable Objects, *in International Conference on Computer Vision and Graphics*, Springer-Verlag, Berlin, Heidelberg, pp.410–419.

Storn, R. and Price, K. V. (1997), Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization* **11**(4), 341–359.

Studholme, C., Little, J. A., Penny, G. P., Hill, D. L. G. and Hawkes, D. J. (1996), Automated multimodality registration using the full affine transformation: Application to MR and CT guided skull base surgery., *in* K. H. Hohne and R. Kikinis (eds.), *LNCS Visualization in Biomedical Computing*, pp.601–606.

Sullivan, J., Blake, A. and Rittscher, J. (2000), Statistical Foreground Modelling for Object Localisation, *Proc. European Conf. Computer Vision* **2**, 307–323.

Sullivan, J., Blake, A., Isard, M. and MacCormick, J. (1999), Object Localization by Bayesian Correlation, *Proc Int. Conf. Computer Vision* pp.1068–1075.

Sullivan, J., Blake, A., M.Isard and J.MacCormick (2001), Bayesian Object Localisation in Images, *Int. J. Computer Vision* **44**(2), 111–136.

Tarr, M. J. and Büllthoff, H. H. (1998), Image-based object recognition in man, monkey and machine, *Cognition* **67**, 1–20.

Tarr, M. J., Williams, P., Hayward, W. and Gauthier, I. (1998), Three dimensional object recognition is viewpoint dependent, *Nature Neuroscience* **1**, 275–277.

Tenenbaum, J. B., de Silva, V. and Langford, J. C. (2000), A Global Geometric Framework for Nonlinear Dimensionality Reduction, *Science* **290**, 2319–2323.

Tian, Q., Yu, J., Xue, Q., Sebe, N. and Huang, T. S. (2004), Robust Error Metric Analysis for Noise Estimation in Image Indexing, *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)* **9**, 140–146.

Tomasi, C. and Kanade, T. (1992), Shape and motion from image streams under orthography., *International Journal of Computer Vision* **9**(2), 137–154.

Tsai, D.-M. and Lin, C.-T. (2003), Fast normalized cross correlation for defect detection, *Pattern Recognition Letters* **24**(15), 2625–2631.

Tsai, D.-M., Lin, C.-T. and Chen, J.-F. (2003), The evaluation of normalized cross correlations for defect detection, *Pattern Recognition Letters* **24**(15), 2525–2535.

Turk, M. and Pentland, A. (1991), Eigenfaces for recognition, *Journal of Cognitive Neuroscience* **3**(1), 71–86.

Ullman, S. (1989), Aligning Pictorial Descriptions: An Approach to Object Recognition, *Cognition* **32**, 193–254.

Ullman, S. and Basri, R. (1991), Recognition by Linear Combinations of Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(10), 992–1006.

Vasanthanayaki, C. and Annadurai, S. (2005), Optimal Morphological Shape Decomposition Scheme, *GVIP* **05**, 1–7.

Vinod, V. and Murase, H. (1996), Object Location Using Complementary Color Features: Histogram And Dct, *Proceedings of the 13th International Conference on Pattern Recognition* **A**, 554–559.

Viola, P. and Wells, W. M. (1995), Alignment by maximization of mutual information, *Proc. 5th Int. Conf. Computer Vision* pp.16–23.

Wiskott, L., Fellous, J.-M., Kruger, N. and von der Malsburg, C. (1999), Face Recognition by Elastic Bunch Graph Matching, *Intelligent Biometric Techniques in Fingerprint and Face Recognition* pp.355–396.

Xu, C. and Prince, J. L. (1998), Snakes, Shapes and Gradient Vector Flow, *IEEE Trans. on Image Processing* pp.359–369.

Xu, Y.-Q., Li, B.-C. and Wang, B. (2004), Face Recognition by Fast Independent Component Analysis and Genetic Algorithm, *in The 4th International Conference CIT*, IEEE Computer Society, Washington, DC, USA, pp.194–198.

Yang, M.-H. (2002), Extended isomap for pattern classification, *in Eighteenth national conference on Artificial intelligence*, Edmonton, Alberta, Canada, pp.224–229.

Yang, M.-H., Kriegman, D. J. and Ahuja, N. (2002), Detecting Faces in Images: A survey, *IEEE Pattern Analysis and Machine Intelligence* **24**(1), 34–58.

Yilmaz, A., Javed, O. and Shah, M. (2006), Object tracking: A survey, *ACM Comput. Surv.* **38**(4), 1–45.

Yoshida, H., Katsuragawa, S., Amit, Y. and Doi, K. (1997), Wavelet-based deformable contour and its application to detection of pulmonarynoduleson chest radiographs, *Proc. of the SPIE* **3169**, 328–336.

Yuille, A. L., Cohen, D. S. and Hallinan, P. (1992), Feature Extraction from Faces Using Deformable Templates, *International Journal of Computer Vision* **8**(2), 99–111.

Zelinka, I. (2004), SOMA-Self Organizing Migrating Algorithm, *in* G. Onwubolu and B. V. Babu (eds.), *New optimization techniques in engineering*, Springer, Berlin.

Zelinka, I. (2006), Investigation on Realtime Deterministic Chaos Control by Means of Evolutionary Algorithms, *in 1st IFAC Conference on Analysis and Control of Chaotic Systems*, Reims, France, pp.28–30.

Zelinka, I. and Nolle, L. (2004), *Differential Evolution - A Practical Approach to Global Optimization*, Springer-Verlag, Chapter Plasma Reactor Optimizing Using Differential Evolution.

Zhou, Q. and Aggarwal, J. K. (2001), Tracking and classifying moving objects from video, *in Proc. of the 2nd IEEE internat. workshop on PETS*, Hawaii.

Zisserman, A. (1992), Notes on Geometric Invariance in Vision, *in BMVC*.

Zografos, V. and Buxton, B. F. (2005a), Affine invariant, model-based object recognition using robust metrics and Bayesian statistics, *in* M. Kamel and A. Campilho (eds.), *Proc. of the Second International Conference on Image Analysis and Recognition (ICIAR 2005)*, Vol. 3656 of *LNCS*, Springer, pp.407–414.

Zografos, V. and Buxton, B. F. (2005b), An evaluation of common distributional models for a Bayesian prior of the scale transformation, Draft.

Zografos, V. and Buxton, B. F. (n.d.), Comparison of optimisation algorithms for deformable template matching., Draft, 2006.